

PERANCANGAN KONTROLER FUZZY 3 DIMENSI UNTUK SUHU, TEKANAN, DAN KELEMBABAN

TUGAS AKHIR

PERPUSTAKAAN ITS	
Tgl. Terima	1 - 8 - 2000
Disusun Oleh : Terima Dari	14
No. Agenda Prp.	21 . 1447

SETIAWARDHANA

NRP : 2295 100 029



JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2000

RSE
629.89
Set
p-1
2000



**PERANCANGAN KONTROLER
FUZZY 3 DIMENSI
UNTUK SUHU, TEKANAN, DAN KELEMBABAN**

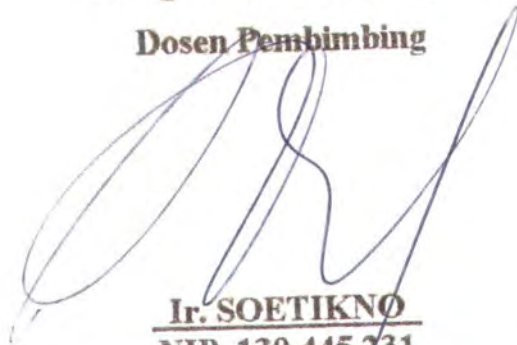
TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro
Pada**

**Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya**

Mengetahui / Menyetujui

Dosen Pembimbing



**Ir. SOETIKNO
NIP. 130 445 231**

**SURABAYA
Pebruari, 2000**



ABSTRAK

Abstrak

Perkembangan elektronika menjadi sangat pesat. Aplikasi dari teknologi elektronika salah satu diantaranya adalah pembuatan alat untuk pengaturan suhu, tekanan, dan kelembaban. Metode logika fuzzy belum optimal apabila kita tidak menggunakan metode yang tepat untuk penyelesaian masalah tersebut.

Pengaturan suhu, tekanan dan kelembaban dapat kita aplikasikan pada alat meratakan gigi (*Orthodontic Appliance*) pada bidang kedokteran gigi.

Pada proses konvensional hanya terukur parameter suhu dan tekanan, sedangkan faktor kelembaban tidak diperhatikan, sehingga pencampuran komposisi bahan yang tidak tepat bisa menyebabkan penurunan mutu hasilnya.

Pada pembuatan alat tugas akhir ini akan dirancang sistem pengaturan suhu, tekanan, dan kelembaban dengan metode *fuzzy logic* berdasar pada logika fuzzy dengan *state machine* tiga dimensi loop tertutup.

Aplikasi konsep dasar *fuzzy logic* tiga dimensi untuk pengaturan suhu, tekanan dan kelembaban dalam suatu ruangan bisa dikembangkan untuk beberapa aplikasi kontrol yang lain dengan *state machine* lebih dari tiga dimensi.



KATA PENGANTAR

KATA PENGANTAR

Dengan mengucapkan syukur Alhamdulillah penulis panjatkan kepada Allah SWT atas segala hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik

Tugas Akhir ini mempunyai beban 4 sks (satuan kredit semester) merupakan salah satu persyaratan untuk memperoleh gelar Sarjana Teknik pada Bidang Studi Elektronika , Jurusan Teknik Elektro , Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam mengerjakan Tugas akhir ini penulis mengacu pada teori Fuzzy Logic yang diperoleh dari literatur dan juga tidak lepas dari arahan Dosen pembimbing yang telah banyak membantu dari awal hingga selesainya pengerjaan Tugas Akhir ini.

Dengan menyadari keterbatasan-keterbatasan yang ada, penyusun menyadari bahwa Tugas Akhir ini masih banyak kekurangan dan masih jauh dari sempurna.

Sebagai akhir kata penyusun berharap semoga Tugas Akhir ini dapat bermanfaat bagi para pembaca sekalian.

Penulis

UCAPAN TERIMA KASIH

Dengan telah selesainya penyusunan Tugas Akhir ini, maka penyusun mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Ir. Soetikno selaku Dosen Pembimbing dan Koordinator Bidang Studi Elektronika yang telah meluangkan waktu dan perhatiannya untuk memberikan saran-saran dan dorongan.
2. Bapak DR.Ir. Achmad Jazidie selaku ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.
3. Eyang kakung , Mama , Om Agus, Om Budi yang telah banyak memberikan dorongan semangat dan doa restunya dalam penyusunan Tugas Akhir ini.
4. Rekan-rekan Bidang Studi Elektronika yang banyak memberi bantuan dan dorongan moral dalam penyusunan Tugas Akhir ini.

Penulis



DAFTAR ISI

DAFTAR ISI

JUDUL i

LEMBAR PENGESAHAN ii

ABSTRAK iii

KATA PENGANTAR iv

UCAPAN TERIMA KASIH v

DAFTAR ISI..... vi

DAFTAR GAMBAR xi

DAFTAR TABEL..... xiii

BAB I PENDAHULUAN..... 1

 1.1 Latar Belakang..... 1

 1.2 Permasalahan 2

 1.3 Pembatasan Masalah..... 2

 1.4 Metodologi..... 2

 1.5 Sistematika Pembahasan..... 3

 1.6 Relevansi..... 3

BAB II TEORI PENUNJANG..... 4

2.1 <i>Fuzzy Logic</i> Tiga Dimensi	4
2.1.1 Teori Dasar <i>Fuzzy Logic</i> Tiga Dimensi	4
2.1.2 State Machine <i>Fuzzy Logic</i> Tiga Dimensi	6
2.2 AL 220 sebagai Kontroller Logika Fuzzy	10
2.2.1 Ciri- ciri dan Aplikasi dari <i>Fuzzy Logic</i> AL 220	10
2.2.2 Penjelasan IC AL 220	12
2.2.3 Arsitektur AL 220	16
2.2.4 Fungsi Keanggotaan	16
2.2.5 Variabel Fuzzy	18
2.2.6 Aturan-aturan	19
2.2.7 Fuzzifikasi dan Defuzzifikasi	20
2.2.8 Organisasi Memori AL 220	21
2.2.9 Pewaktuan	28
2.2.10 Tundaan <i>Internal Loopback</i>	29
2.2.11 Operasi <i>Prescale</i>	30
2.2.12 Rangkaian <i>Clock</i> dan <i>Power Up Reset</i>	30
2.2.13 Rangkaian <i>Power Reset</i>	31
2.2.14 Diagram Waktu untuk AL 220	31
2.3 <i>Analog To Digital Converter (ADC)</i>	32
2.4 <i>Digital To Analog Converter (DAC)</i>	34
2.5 Mikrokontroler 80C31BH	35

BAB III PERENCANAAN DAN PEMBUATAN PERANGKAT KERAS DAN	
PERANGKAT LUNAK.....	37
3.1 Perancangan Keseluruhan Sistem	37
3.1.1 Rangkaian Sensor Suhu dan Pengkondisi Sinyalnya	38
3.1.2 Rangkaian Sensor Kelembaban dan Pengkondisi Sinyalnya	40
3.1.3 Rangkaian Sensor Tekanan dan Pengkondisi Sinyalnya.....	43
3.1.4 Rangkaian <i>Analog To Digital Converter</i>	44
3.1.5 Rangkaian <i>Digital To Analog Converter</i>	45
3.1.6 Rangkaian Mikrokontroler 80C31BH	46
3.2 Rangkaian <i>Fuzzy Logic</i> AL 220.....	47
3.3 Rangkaian Aktuator	47
3.3.1 Rangkaian Aktuator untuk Plant Suhu	48
3.3.1.1 Rangkaian Komparator	48
3.3.1.2 Rangkaian <i>Pulse Width Modulation</i>	48
3.3.1.3 Rangkaian Opto Triac	51
3.3.2 Rangkaian Aktuator untuk Plant Kelembaban	52
3.3.3 Rangkaian Aktuator untuk Plant Tekanan.....	53
3.4 Perancangan Perangkat Lunak <i>Fuzzy Logic</i> Tiga Dimensi	54
3.4.1 Teori Suhu, Tekanan dan Kelembaban	54
3.4.2 <i>Heuristic</i> Perangkat Lunak Fuzzy Tiga Dimensi	61
3.4.2.1 Perancangan Fungsi Keanggotaan untuk Suhu	61

3.4.2.2 Perancangan Fungsi Keanggotaan untuk Kelembaban	67
3.4.2.3 Perancangan Fungsi Keanggotaan untuk Tekanan.....	68
3.4.2.4 Penjelasan <i>Rule-rule</i> Perangkat Lunak <i>Fuzzy Logic</i>	70
3.5 Perancangan Perangkat Lunak untuk Supervisi.....	74
 BAB IV PENGUJIAN DAN PENGUKURAN	75
4.1 Pengujian Alat	75
4.2 Kalibrasi dan Pengukuran	76
4.2.1 Suhu	76
4.2.2 Kelembaban	77
4.2.3 Tekanan	78
4.2.4 Modul Supervisi yang terdiri dari ADC dan DAC	78
4.3 Pengujian Modul <i>Fuzzy Logic</i> AL 220	80
4.3.1 Pengujian dan pengukuran pada respon AL 220	76
4.3.2 Perbandingan antara hasil rancangan dengan kompilasi	77
 BAB V PENUTUP	83
5.1 Kesimpulan	83
5.2 Saran.....	84
 DAFTAR PUSTAKA	85

LAMPIRAN87

LAMPIRAN A : Perangkat Lunak *Fuzzy Logic* F.A.E 3.087

LAMPIRAN B : Skema Rangkaian97

LAMPIRAN C : Perangkat Lunak Mikrokontroller104

LAMPIRAN D : *Data Sheet* AL 220125



DAFTAR GAMBAR

DAFTAR GAMBAR

Gambar 2.1 Diagram Blok Sistem Tiga Dimensi	5
Gambar 2.2 State Machine <i>Fuzzy Logic</i> 3 Dimensi	9
Gambar 2.3 IC AL 220	12
Gambar 2.4 Diagram Blok AL 220	16
Gambar 2.5 Tipe Fungi Keanggotaan AL 220	17
Gambar 2.6 Fungi Keanggotaan Bentuk <i>Trapezoidal</i>	18
Gambar 2.7 Metode <i>Immediate</i> dan <i>Accumulate</i>	21
Gambar 2.8 Rangkaian <i>Clock</i>	30
Gambar 2.9 Rangkaian <i>Reset</i>	31
Gambar 2.10 Diagram Waktu dari AL 220	31
Gambar 2.11 Diagram Blok ADC MAX 158	33
Gambar 2.12. Kontrol Layar LCD	36
Gambar 3.1 Diagram Blok Sistem Tiga Dimensi	38
Gambar 3.2 Pengkondisi Sinyal untuk Suhu	39
Gambar 3.3 Pengkondisi Sinyal untuk Kelembaban	42
Gambar 3.4 Pengkondisi Sinyal untuk Tekanan	44
Gambar 3.5 Rangkaian ADC MAX 158	45
Gambar 3.6 Rangkaian DAC MAX 505	45

Gambar 3.7 Rangkaian AL 220	47
Gambar 3.8 Diagram Blok Aktuator Plant Suhu.....	48
Gambar 3.9 Rangkaian Komparator	49
Gambar 3.10 Rangkaian <i>PWM</i>	50
Gambar 3.11 <i>PWM</i> dengan keluaran Invertong	50
Gambar 3.12 Rangkaian Opto Triac	51
Gambar 3.13 Rangkaian Aktuator untuk Plant Kelembaban	52
Gambar 3.14 Rangkaian Aktuator untuk Plant Tekanan	53
Gambar 3.15 Variabel Fungsi Keanggotaan	61
Gambar 3.16 Fungsi Keanggotaan Dimensi X untuk Suhu	64
Gambar 3.17 Fungsi Keanggotaan untuk <i>Error</i>	66
Gambar 3.18 Fungsi Keanggotaan Dimensi Y untuk Kelembaban	68
Gambar 3.19 Fungsi Keanggotaan Dimensi Z untuk Tekanan	69
Gambar 4.1 Grafik Pengujian Sistem.....	82
Gambar 4.2 Grafik Pengujian Tegangan <i>Fuzzy Logic</i>	82



DAFTAR TABEL

DAFTAR TABEL

Tabel 2.1 Tabel <i>Absolute Rating</i> TA = 25 °C	14
Tabel 2.2 Tabel Spesifikasi Konversi Analog.....	14
Tabel 2.3 Tabel Spesifikasi dan Kondisi Operasi yang diijinkan	15
Tabel 2.4 Tabel Organisasi Memori.....	22
Tabel 2.5 Tabel <i>Command Byte (Even Addresses)</i>	23
Tabel 2.6 Tabel <i>Select Byte (Odd Addresses)</i>	26
Tabel 2.7 Tabel <i>Clock Circuit</i>	30
Tabel 2.8 Tabel Kebenaran untuk selektor data masukan A0-A2	33
Tabel 2.9 Pengalamatan DAC MAX 505 (partial list)	35
Tabel 3.1 Tabel Teori <i>MIN</i>	62
Tabel 3.2 Tabel Teori <i>MAX</i>	63
Tabel 3.3 Tabel Teori <i>NOT</i>	63
Tabel 4.1 Tabel Pengukuran Suhu	77
Tabel 4.2 Tabel Pengukuran Kelembaban	77
Tabel 4.3 Tabel Pengujian Modul ADC	79
Tabel 4.4 Tabel Pengujian Modul DAC	79
Tabel 4.5 Tabel Pembacaan Sistem hasil aksi <i>Fuzzy Logic</i>	81

Tabel 4.6 Tabel Pembacaan Sistem dengan alat kalibrasi	81
Tabel 4.7 Tabel Pembacaan Keluaran <i>Fuzzy Logic</i>	81
Tabel 4.8 Tabel Error Sistem	81



BAB I
PENDAHULUAN

BAB I

PENDAHULUAN

Pada bab ini akan dibahas mengenai latar belakang, permasalahan, pembatasan masalah, metodologi, sistematika pembahasan, dan relevansi dari tugas akhir yang dibuat.

1.1 Latar Belakang

Pada era globalisasi ini, perkembangan elektronika menjadi sangat pesat. Aplikasi dari elektronika ini sekarang sudah dapat diterapkan pada hampir seluruh bidang kehidupan kita. Aplikasi elektronika di bidang kontrol dengan teknologi *fuzzy logic* untuk *fuzzy logic* tiga dimensi dengan tiga parameter kontrol dengan tiga aksi kontrol.

Pada pembuatan alat tugas akhir ini akan dirancang suatu sistem pengaturan *fuzzy logic* tiga dimensi dengan *fuzzy logic* IC AL 220 dengan menggunakan konsep kontrol loop tertutup berumpan balik.

Beberapa keunggulan teknologi *fuzzy logic* dibanding dengan teknologi logika biasa adalah lebih handal, lebih halus, dan lebih cepat dibanding dengan teknologi konvensional biasa.

Pada tugas akhir ini akan dirancang dengan *fuzzy logic* tiga dimensi dengan prinsip *state machine* , *close loop* , dan diaplikasikan untuk pengaturan suhu, tekanan dan kelembaban dalam suatu ruangan.

1.2 Permasalahan

Permasalahan dalam tugas akhir ini adalah perancangan dan pembuatan sistem untuk pengaturan suhu, tekanan, dan kelembaban dengan prinsip *fuzzy logic* tiga dimensi dengan IC AL 220, dengan kontrol berumpan balik dan loop tertutup bisa dicapai kondisi mendekati ideal.

1.3 Pembatasan Masalah

Pembatasan masalah dalam tugas akhir ini adalah perancangan sistem untuk pengaturan suhu, tekanan, kelembaban dengan prinsip *fuzzy logic* tiga dimensi dengan menggunakan IC AL 220.

1.4 Metodologi

Untuk memecahkan permasalahan yang ada, maka dalam pengerjaan tugas akhir ini dilakukan langkah-langkah sebagai berikut:

1. Studi literatur mengenai konsep *fuzzy logic*
2. Melakukan penelitian terhadap karakteristik *fuzzy logic* AL220
3. Melakukan perancangan dan pembuatan rangkaian, meliputi perancangan dan pembuatan perangkat keras dan perangkat lunak. Setelah itu dilakukan pengujian terhadap masing-masing bagian dan secara keseluruhan.
4. Perancangan dan pembuatan perangkat keras dan perangkat lunak untuk *fuzzy logic* AL 220.
5. Secara keseluruhan langkah diatas disusun dalam suatu laporan tugas akhir.

1.5 Sistematika Pembahasan

Sistematika pembahasan di dalam tugas akhir disusun sebagai berikut:

- BAB I berisi pendahuluan yaitu tentang latar belakang, permasalahan, pembatasan permasalahan, metodologi, sistematika pembahasan, dan relevansinya.
- BAB II berisi teori penunjang *fuzzy logic* tiga dimensi dan teori dasar *fuzzy logic* AL 220.
- BAB III membahas perencanaan dan pembuatan perangkat keras dan perangkat lunak untuk *fuzzy logic* AL 220 beserta rangkaian transducer, pengkondisi sinyal, mikrokontroler 80C31BH sebagai perangkat keras *supervision*, dan sistem akuisi data yaitu ADC dan DAC yang digunakan.
- BAB IV berisi tentang pengujian dan pengukuran sistem, baik secara perbagian maupun keseluruhan.
- BAB V merupakan bagian penutup yang berisi kesimpulan dan saran.

1.6 Relevansi

Dengan dibuatnya alat ini diharapkan dapat diterapkan pada aplikasi kontrol suhu, tekanan, dan kelembaban dengan konsep *fuzzy logic* tiga dimensi.



BAB II
TEORI PENUNJANG

BAB II

TEORI PENUNJANG

Pada bab ini akan dibahas mengenai teori dasar dari *fuzzy logic* tiga dimensi dan teori dasar *fuzzy logic* AL 220.

2.1. *Fuzzy logic* tiga dimensi

Fuzzy logic tiga dimensi merupakan salah satu metode penyelesaian masalah untuk pengaturan suhu, tekanan, dan kelembaban dengan logika fuzzy.

2.1.1. Teori dasar *fuzzy logic* tiga dimensi

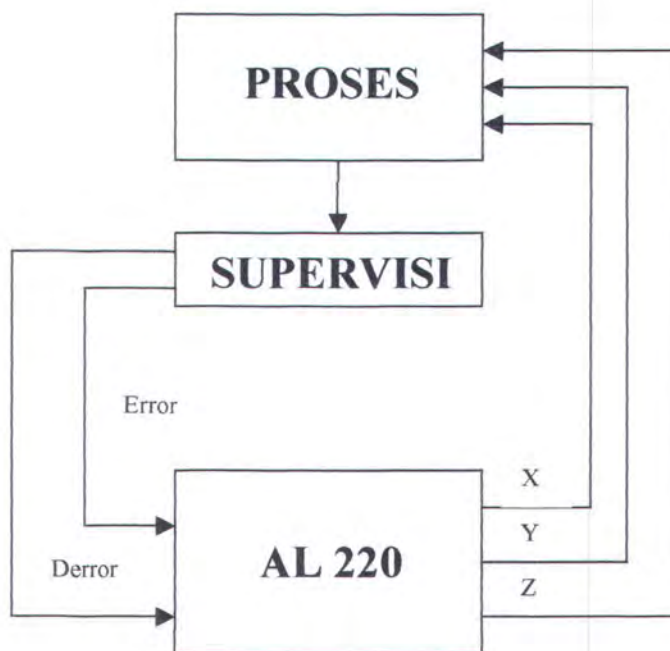
Fuzzy logic tiga dimensi adalah suatu prinsip yang berdasar pada prinsip *multidimensional convergence*¹ dimana ada banyak dimensi yang bertemu untuk menghasilkan suatu aksi kontrol tertentu. Pada beberapa aplikasi sistem kontrol sering dibutuhkan untuk mengatur dua atau lebih sinyal yang berubah secara dinamik untuk mencapai setting yang optimum.

Fuzzy logic tiga dimensi x, y dan z sebagai koordinat dari suhu, kelembaban, dan tekanan, berubah – ubah secara dinamik untuk mencapai kestabilan sistem, sehingga diselesaikan dengan beberapa model, dikembangkan untuk kontrol adaptif bersifat *self-adjustment*, seperti sistem adaptif, neural kontrol, dan sistem hibrida.

¹ Adaptive Logic , AL220 Stand Alone Fuzzy logic Controller Data Sheet, Basicconcepts , Inc

Ada 2 macam model yaitu *supervised learning* dan *unsupervised learning*. Pada tugas akhir ini di kembangkan sistem dengan *supervised learning*, *fuzzy logic* tiga dimensi, dengan dibutuhkan suatu pembentukan *state machine* tiga dimensi.

Ilustrasi dari system dengan *fuzzy logic* tiga dimensi dengan diagram blok sistem untuk mengontrol sinyal yang tidak linier dan dinamik adalah

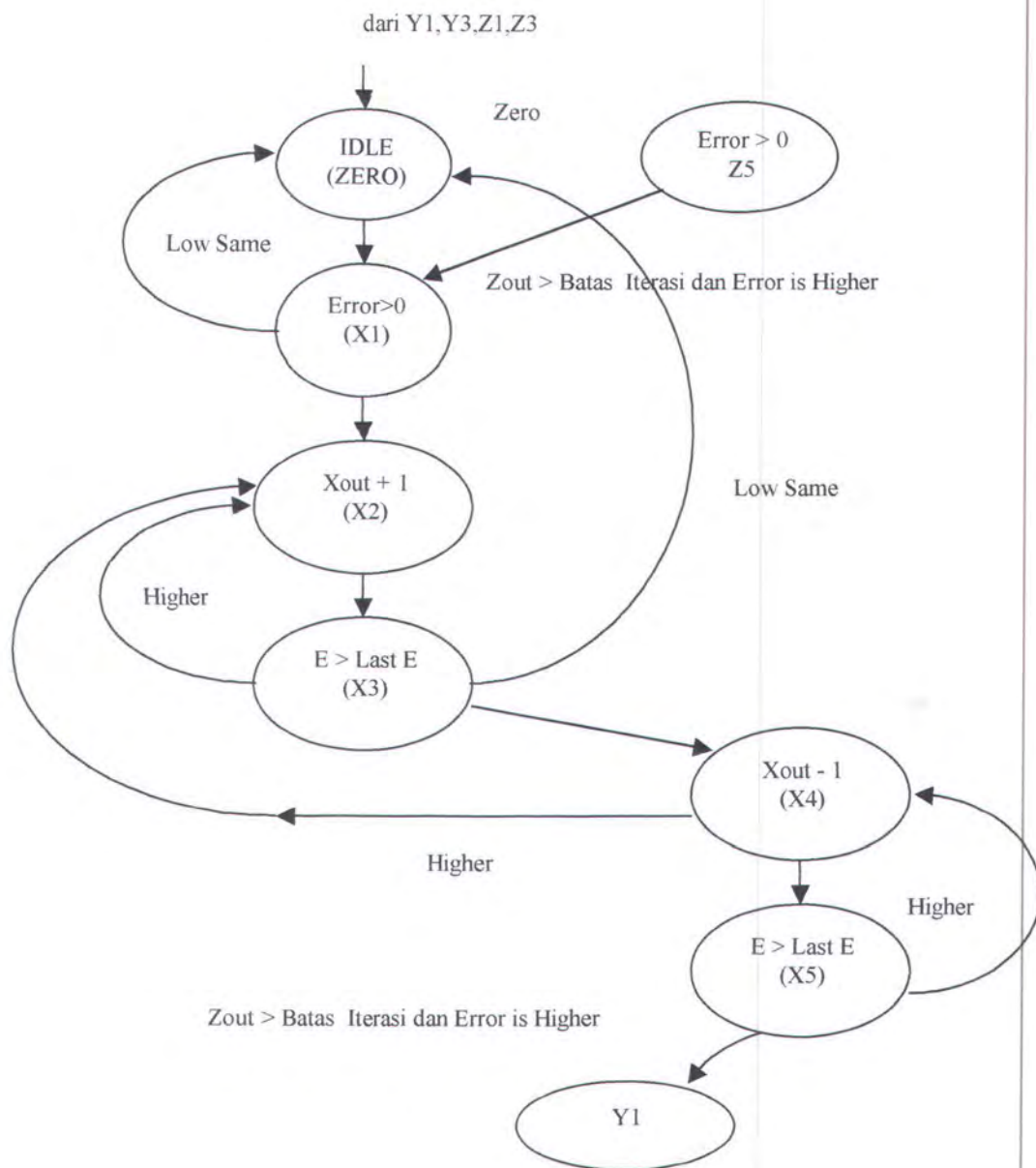


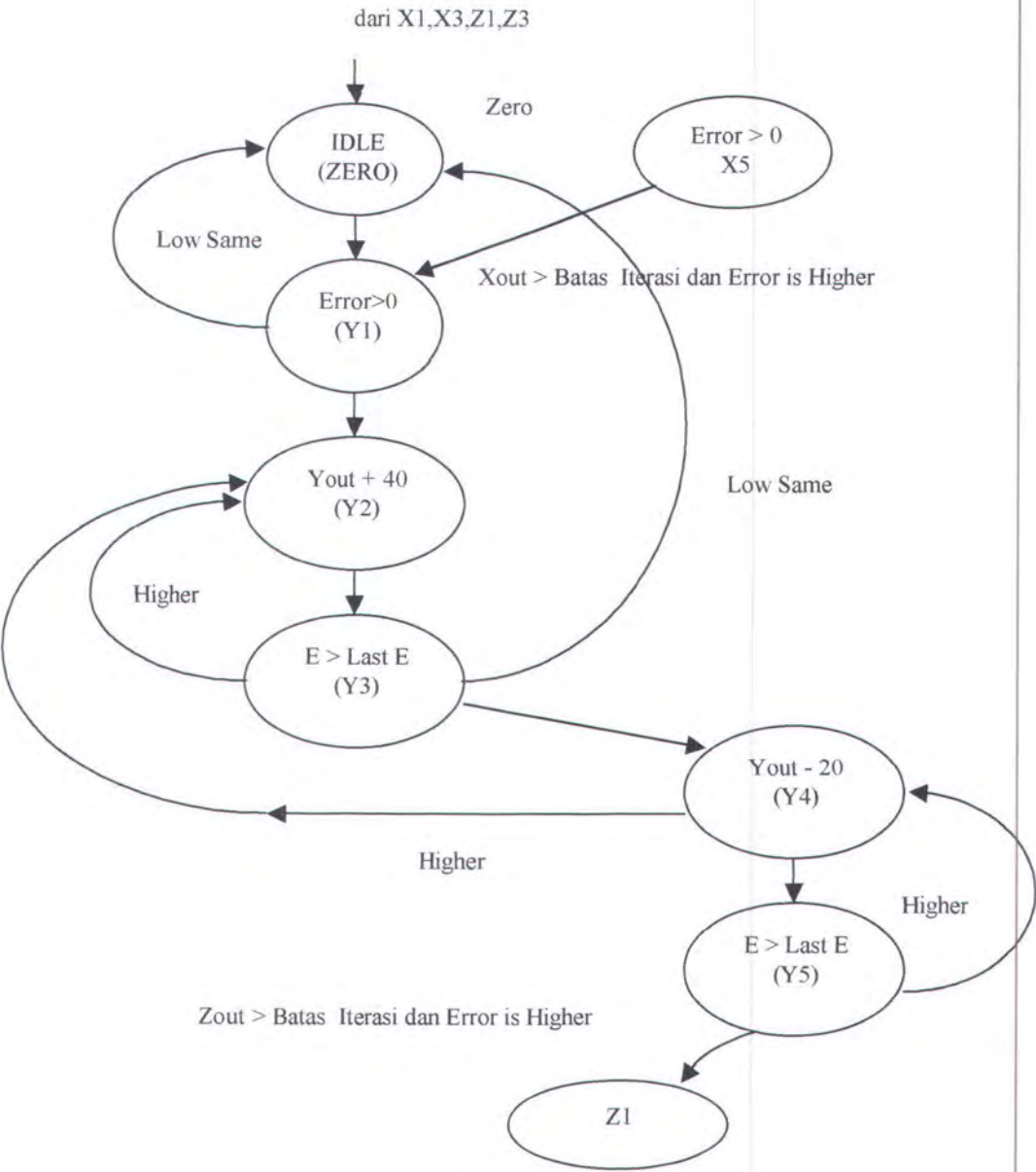
Gambar 2.1. Diagram blok sistem tiga dimensi

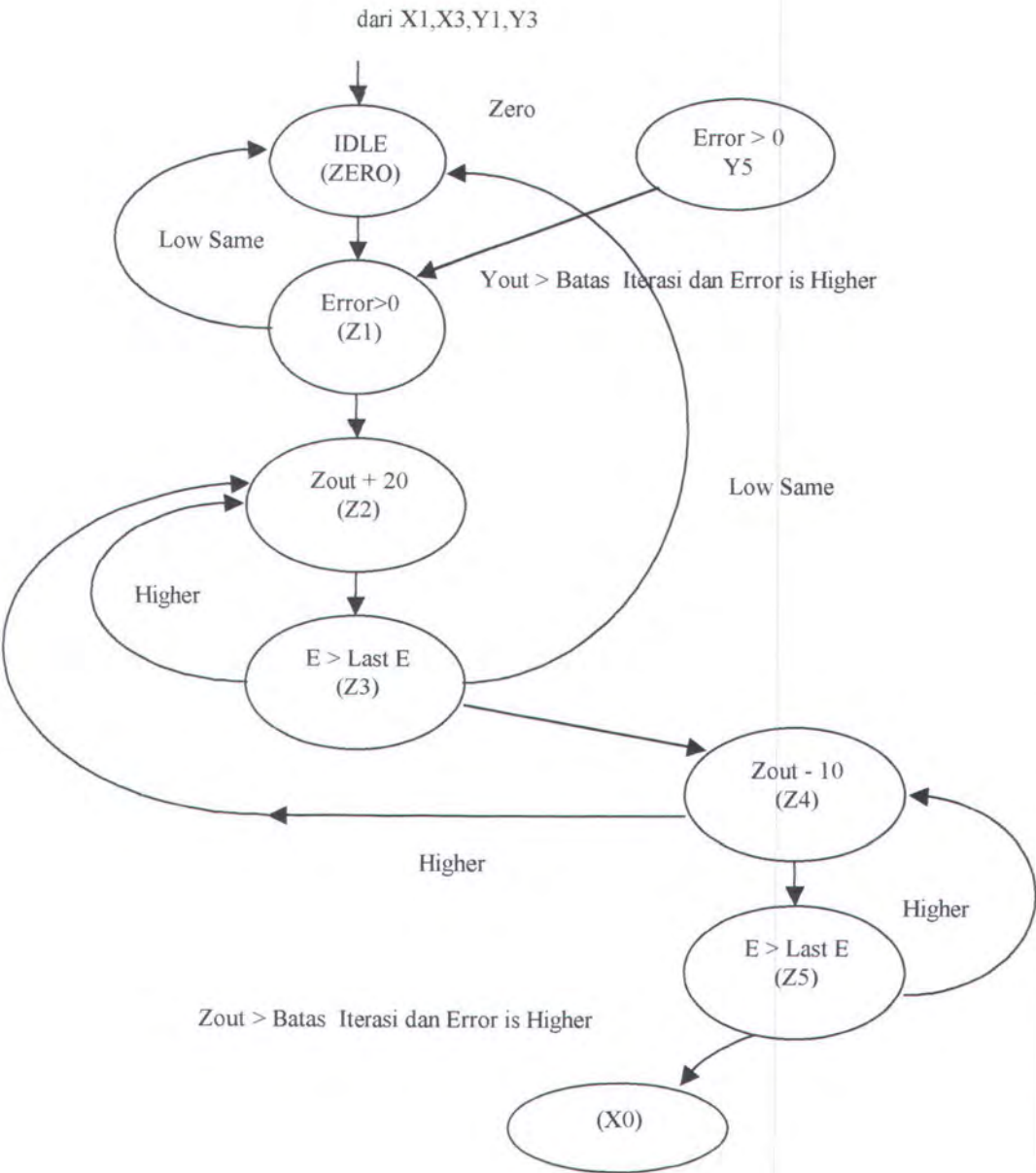
2.1.2. State machine fuzzy logic tiga dimensi

State machine fuzzy logic tiga dimensi ² dapat dilihat pada gambar 2.2 bahwa sinyal masukan pada fuzzy AL 220 adalah berupa dua buah sinyal yaitu sinyal *error* dan *D error* dimana ada pembandingan pada AL 220 apakah sinyal *error* tersebut bertambah atau berkurang dengan menggunakan *floating membership function*. State berawal pada state zero, dimana sistem yang dirancang bersifat loop tertutup, dari x ke y ke z kemudian *loop back*. Pada state zero ini sistem akan *idle* apabila sistem mendeteksi masukan *error* dan *D error* sama dengan nol. Kemudian dari state zero ke state X1 distate ini akan di bandingkan apakah *error* lebih besar dari nol atau lebih kecil sama dengan nol, apabila sistem lebih kecil sama dengan nol maka sistem akan *idle*, berarti tidak ada *error* atau kondisi setting telah terpenuhi. State X2 akan menambah Xout untuk di bandingkan bila *error* lebih besar nol dengan *D error*, bila penambahan tersebut *error* lebih kecil sama dengan *D error* maka di koreksi apakah *error* lebih besar dari nol, bila *error* masih besar maka kembali ke state X2, bila *error* lebih kecil sama dengan *D error* maka mengurangi Xout di state X4 kemudian di koreksi di state X5 apakah *error* lebih besar dibandingkan dengan *D error*, bila ya maka kembali ke state X2, bila memang lebih kecil sama dengan maka berarti aksi yang dilakukan benar dan kembali ke state X4, kemudian state X5 juga mengkoreksi apakah *error* terlalu besar, bila terlalu besar dan batas iterasi maksimum maka state koreksi pada state Y1 dimana proses pada State Y sama dengan pada State X, begitu juga untuk aksi Z

² *ibid.*, p.8







Gambar 2.2. State machine fuzzy logic 3 dimensi

2.2. AL220 sebagai kontroller logika fuzzy

Pada pertengahan tahun 1960-an Proff. Lotfi Zadeh dari University of California at Berkeley mengemukakan bahwa prinsip sifat benar atau salah tidak dapat menerangkan banyak kejadian di alam semesta, dimana merupakan gradasi dari benar dan salah.

Untuk menerangkan gradasi tak tentu dari benar dan salah dikemukakan teori himpunan klasik yang disebut himpunan fuzzy, tidak seperti logika boole, logika fuzzy merupakan logika dengan nilai banyak, sehingga logika fuzzy berhubungan dengan derajat keanggotaan (*degrees of membership*) dan derajat kebenaran (*degrees of truth*), sehingga sesuatu dapat bernilai benar dan salah dalam waktu tertentu³, misal untuk suhu 50 °C apakah tergolong panas atau hangat, maka dalam dunia nyata semuanya tergolong benar, sehingga diperoleh bahwa dalam logika boole dapat menyebabkan reaksi sistem yang berbeda sedangkan dalam logika fuzzy perubahan kecil dari 50 °C ke 50,1 °C hanya pergeseran dalam sistem tersebut.

³ Jamshidi Mohammad dan Vadiie, Nader dan Ross, Timothy J., Fuzzy logic and Control, Software dan Hardware Applications, Prentice Hall 1993

2.2.1. Ciri-ciri dan aplikasi dari *fuzzy logic* AL220

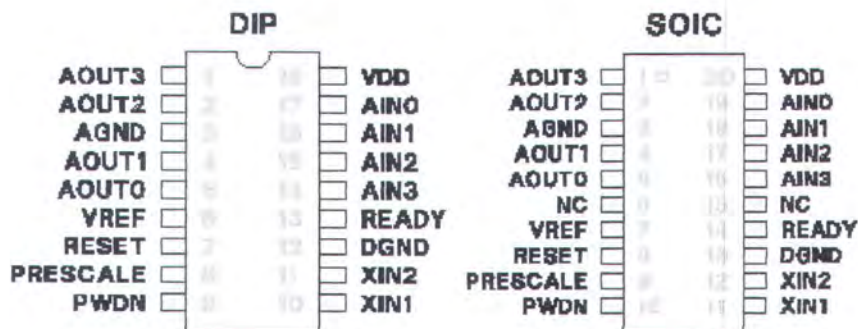
Ciri - ciri dari AL220 sebagai IC *fuzzy logic* kontroller yang bersifat *stand alone* adalah sebagai berikut :

1. *Single Chip Fuzzy logic processor.*
2. *Self-Adaptif Control.*
3. Mempunyai *internal EEPROM* terprogram.
4. 8 - bit masukan analog dan 8 - bit keluaran analog.
5. 6 tipe fungsi keanggotaan.
6. 111 Variabel fuzzy dan lebih dari 50 rule.
7. Sistem dirancang berbasis komputer.
8. Kemasan 18 pin DIP atau 20 pin SOIC.

Aplikasi dari AL 220 adalah :

1. Aplikasi cerdas.
2. Otomotif.
3. Kontrol motor.
4. Management Baterai.
5. Sistem Komunikasi.

2.2.2. Penjelasan IC AL220



Gambar 2.3. IC AL220

Gambaran umum dari kaki IC AL 220 sebagai berikut :

□ Masukan

1. Reset

Reset aktif dengan sinyal clock aktif *low* untuk inisialisasi perangkat keras dan aktif minimal 8 *clock cycle* untuk memastikan operasi yang baik. Dapat dioperasikan dengan *power-up delayed circuit* artinya dalam keadaan aktif tinggi AL 220 berada dalam mode daya rendah .

2. Ain(3:0)

Masukan data analog dikonversi ke 8 bit data digital. Masukan analog yang tidak digunakan seharusnya terhubung ke ground.

3. Xin

Clock masukan di gerakkan dari *external clock*, bisa dari *internal clock* dengan pemasangan R dan C secara eksternal.

4. Prescale

Logika "1" membuat perangkat keras aktif *prescale* dan logika "0" membuat perangkat keras aktif normal atau *prescale* tidak digunakan, atau dihubungkan ke kaki ready jika dipakai terus menerus, Mode dapat disiapkan selama operasi. Setelah "Reset" kondisi prescale harus *low* selama *4 clock cycles*.

□ Keluaran

1. Aout(3:0)

Keluaran data analog, dari 8 bit data digital di konversi ke level analog secara internal.

2. Ready

Setelah reset kaki ini memberikan tanda start bahwa alat mulai untuk mensampling dan memproses data, kaki ini harus dibiarkan dihubungkan ke *prescale*.

3. Vref

Memfilter internal reference voltage, dihubungkan ke ground dengan kapasitor 0.1 uF.

Karakteristik dari AL220 dapat di tabelkan pada tabel 1 , tabel 2 dan tabel 3 sebagai berikut :

Tabel 2.1. Tabel Absolute Maximum Rating TA=25°C

Parameter	Min	Max	Units
VDD TO GND	-0.5	7.0	V
VSS TO GND	0	0	V
DIGITAL INPUT	0	VDD	V
ANALOG INPUT	0	VDD	V
POWER DISSIPATION		100	mW
STORAGE TEMPERATURE	-50	150	°C

Tabel 2.2. Tabel Spesifikasi konversi analog

Parameter	Value	Units
Resolution	1	BIT
Zero Code Error	1±	LSB
Full Scale Error	1±	LSB
SNR	45	DB Min
Slew Rate, Tracking	1.6	V/mS max
Sampling Rate	10kHz	Per Channel

Tabel 2.3. Spesifikasi dan kondisi operasi yang diijinkan

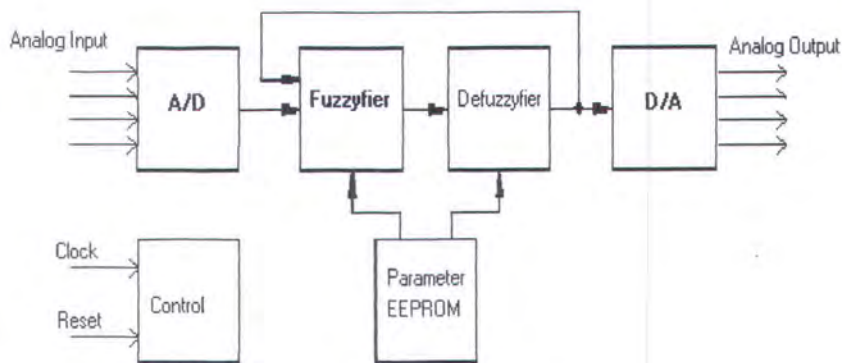
Parameter	Min	Norm	Max	Units
VDD Supply Voltage	4.75	5.0	5.25	V
IDD Supply Current				mA
IoL Digital Output Low-Level Current			5	mA
IoH Digital Output High-Level Current			-5	mA

Parameter	Min	Norm	Max	Units
F Clock Frequency	1		10	MHz
ViL Digital input Low Level voltage	0		0.8	V
ViH Digital input High Level voltage	3.5		Vdd	V
IiL Digital input Low Level current			-40	μA
IiH Digital input Low Level current			1	μA
Zin Analog input impedance	100	150	250	Kohm
Vin Analog input voltage range	0		Vdd-0.5	V
Vo Analog output voltage range	Vss+0.5		Vdd-0.5	V
Io Analog output current	-5		5	mA
Tw Reset pulse width	100			mS
Tsv Reset inactive before clock	10			mS
TA Operating Ambient Temperature	0		70	°C

2.2.3. Arsitektur AL 220

AL 220 sebagai *stand-alone fuzzy logic* dengan operasi kontrol dirancang untuk aplikasi dengan rule yang sederhana. AL 220 terdiri dari element utama ⁴ yaitu *fuzzyfier*, *defuzzyfier*, dan *controller unit*.

Fuzzifikasi berfungsi untuk mengkonversi data dari masukan data analog ke data fuzzy. Fuzzifikasi mengevaluasi data fuzzy sesuai dengan definisi *rule - rule* sesuai dengan sistem yang dikontrol. Setelah rule-rule dievaluasi maka defuzzifikasi memberi tanda harga ke keluaran.

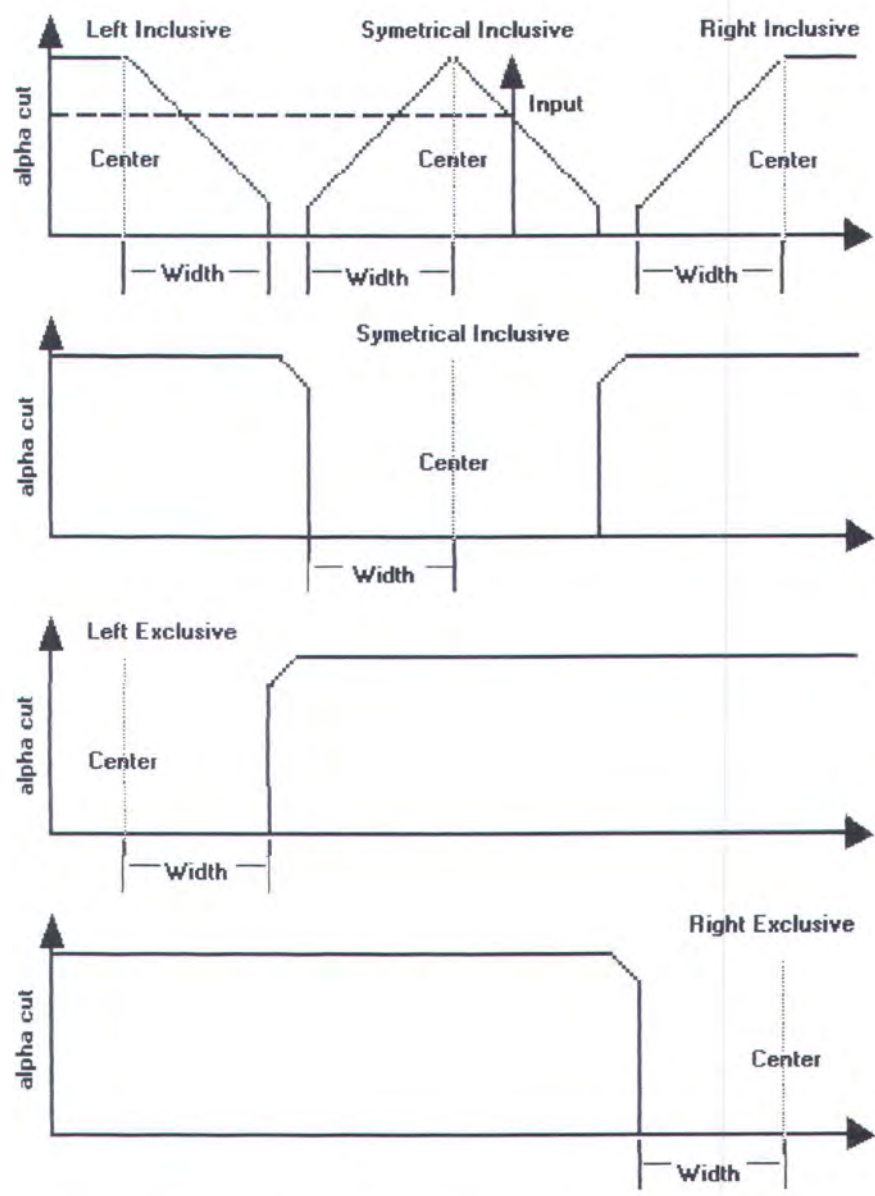


Gambar 2.4. Diagram Blok AL 220

2.2.4. Fungsi keanggotaan

Fungsi keanggotaan merupakan definisi dari variasi input untuk menghasilkan keluaran yang akan dikontrol dengan proses perhitungan dengan *fuzzy logic*. Berikut adalah beberapa tipe fungsi keanggotaan :

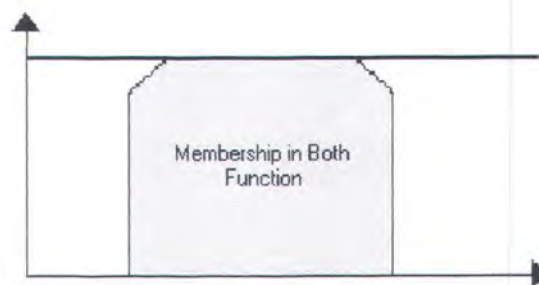
⁴ Adaptive Logic , AL220 Stand Alone Fuzzy logic Controller Data Sheet, Basicconcepts , Inc



Gambar 2.5. Tipe fungsi keanggotaan AL 220

Fungsi keanggotaan berfungsi untuk memilah- milah perubahan masukan untuk dibandingkan antara data masukan dengan logika fuzzy secara langsung sehingga dapat mengubah nilai pusat dan lebar secara bersamaan dengan berubahnya data masukan atau keluarannya.

AL 220 mendukung untuk enam macam fungsi keanggotaan, kemiringan konstan (gradien tetap) , terdiri dari fungsi *Left*, *Symetrical* dan *Right Inclusive* dan inversenya yaitu *exclusive* yang dapat diatur lebarnya Kita juga dapat membuat fungsi keanggotaan yang overlap antara *left* dan *right inclusive* fungsi keanggotaan membentuk pola trapezoidal.



Gambar 2.6. Fungsi keanggotaan bentuk trapezoidal

2.2.5. Variabel Fuzzy

Variabel fuzzy menunjukkan assosiasi masukan untuk expresi tiap masukan , berada pada sumbu axis dengan referensi fungsi keanggotaan dan variabel masukan. Dengan contoh sebagai berikut :

If Error is Zero

" Error " mengacu pada masukan dan " Zero " mengacu pada fungsi keanggotaan. Asosiasi di bentuk dengan fuzzifikasi. Hasil dari variabel fuzzy adalah kecocokan dengan fungsi keanggotaan. Pada AL 220 harga dari data-data fuzzy antara range 0 sampai 63.

2.2.6. Aturan – aturan

Aturan- aturan atau *rule* terdiri dari satu atau lebih variabel fuzzy dan harga aksi keluarannya. Aturan-aturan tersebut memberitahukan controller untuk merespon perubahan masukan data.

If Error is Zero then Temperature +0

" Temperature + 0 " menunjukkan aksi *floating* penambahan nilai integral dengan nilai sebelumnya yaitu accumulate.

Evaluasi *rule* terdiri dari 2 step teknik *MAX-of-MIN*, dimana pada step pertama (*MIN*) semua harga pada variabel fuzzy di *rule*-nya dibandingkan. Harga terendah dominan pada *rule*, pada step kedua (*MAX*) harga semua variabel *rule* dibandingkan dengan harga tertinggi dominan.

Fungsi keanggotaan *floating* pada AL 220 mempunyai pusat dan lebar yang dinamik yang tepat disimpan pada memory. Fungsi keanggotaan *floating* dengan variabel fuzzy dengan pengukuran yang tepat pada perbedaan pengukuran dua masukan. Koreksi berguna untuk mengatur *floating center* dari fungsi keanggotaan dalam *rule* untuk pemrosesan data.

2.2.7. Fuzzifikasi dan Defuzzifikasi

Proses pada AL 220 terdiri dari beberapa tahap⁵ yaitu :

- ❑ Mengambil sampel data analog yang dikonversi ke data digital.
- ❑ Fuzzifier membandingkan data digital dengan variabel fuzzy.
- ❑ Bentuk fuzzifikasi adalah *MAX-of-MIN* untuk penentuan *rule* yang menang / dimunculkan pada keluarannya
- ❑ Defuzzifikasi mengambil yang menang dan diambil untuk di konversikan ke keluaran analog atau ke internal feedback.

Proses Fuzzifikasi

Proses fuzzifikasi dilakukan dengan cara mengambil sampel data analog yang dikonversi ke data digital , fuzzifikasi membandingkan data digital dengan variabel fuzzy. Bentuk fuzzifikasi adalah *MAX-of-MIN* untuk penentuan *rule* yang menang / dimunculkan. Defuzzifikasi mengambil yang menang dan diambil untuk di konversikan ke keluaran analog atau ke *internal feedback*. *Rule* dievaluasi oleh setting *rule* atau referensi keluaran yang lainnya. Keluaran akan berubah setiap satu kali siklus pada setiap masukan data yang berubah

⁵ Adaptive Logic, AL220 Stand Alone Fuzzy logic Controller Data Sheet, Basicconcepts , Inc. ,p.7

Proses Defuzzifikasi

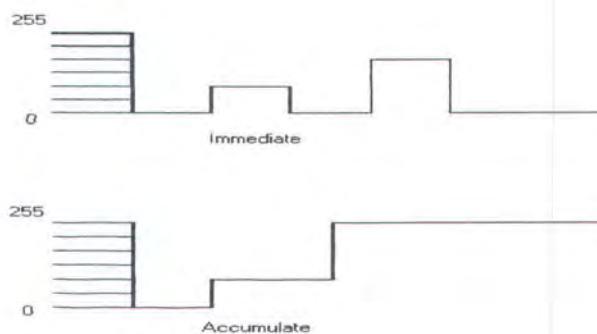
Defuzzifikasi mengambil yang *rule* menang untuk di konversikan ke t analog atau ke *internal feedback*.

Metode Defuzzifikasi

Metode *Defuzzifikasi* menyebabkan beberapa aksi *rule* yang menang untuk kontrol keluaran . AL 220 mempunyai dua metode defuzzifikasi yaitu *immediate* dan *Accumulate* .

- ❑ Metode *Immediate* : Fungsinya seperti " lookup table " dimana aksinya langsung ke keluaran
- ❑ Metode *Accumulate* : Penambahan atau pengurangan keluaran sekarang dengan memperhatikan aksi keluaran sebelumnya.

2.2.8. Organisasi Memori AL 220



Gambar 2.7. Metode *Immediate* dan *Accumulate*⁶

⁶ Ibid. , p.7

AL220 mempunyai *EEPROM* 256-byte untuk menyimpan parameter dari aplikasi yang terdiri dari 32 byte untuk menyimpan fungsi keanggotaan *fixed* untuk harga pusat dan lebarnya. Sisanya 224 di organisasi untuk satu atau lebih *rule-rule* dengan satu atau lebih variabel fuzzy tiap *rule*-nya. Tiap *rule*-nya sedikitnya membutuhkan 2 byte ditambah sedikitnya 2 byte untuk variabel fuzzy tiap rulanya. Memori dibagi dalam tiga bagian yaitu *rule/variabel storage* , *center storage* dan *width storage*. Organisasi memori didefinisikan sebagai berikut :

Tabel 2.4. Tabel Organisasi Memori

Dec Address	Hex Address	Function
0	00	Rules
<>	<>	<>
223	DF	Rules
224	E0	Centers
<>	<>	<>
239	EF	Centers
240	F0	Widths
<>		<>
255	FF	Widths



Rule dan penyimpannya dapat kita analisa sesuai dengan tabel sebagai berikut:

Tabel 2.5. Tabel *Command Byte (Even Addresses)*

7	6	5	4	3	2	1	0
WF	CF	I/O CONT	I/O SELECT		TYPE 2-7		
AF	MODE				TYPE 1		
AF	MODE		OUTPUT SELECT		TYPE 0		

Kunci :

Type	<u>210</u>	
	000	Last Term of Last Rule given output
	001	Last Term of Current Rule
	010	MF, Symetrical Inclusive
	011	MF, Symetrical Exclusive
	100	MF, Left Inclusive
	101	MF, Left Exclusive
	110	MF, Right Inclusive
	111	MF, Right Exclusive

I/O SELECT	<u>43</u>	
	00	I/O PORT 0 AS INPUT
	01	I/O PORT 1 AS INPUT
	10	I/O PORT 2 AS INPUT
	11	I/O PORT 3 AS INPUT
I/O CONTROL	<u>5</u>	
	0	SELECT FROM INPUTS
	1	SELECT FROM OUTPUTS
MODE	<u>6</u>	
	0	Immediate output equals ACTION
	1	Accumulate output equals current output plus 2's complement action (-128 to + 127)
AF	<u>7</u>	
	0	Select Action From Select Byte (fixed)
	1	Select Action From I.O Via Select Byte (float)

Output Select	<u>43</u>	
	00	ACTION from current RULE set to Output 0
	01	ACTION from current RULE set to Output 1
	10	ACTION from current RULE set to Output 2
	11	ACTION from current RULE set to Output 3
CF	<u>6</u>	
	0	Select center from memory via select byte (fixed)
	1	Select center from I/O via select byte (float)
WF	<u>7</u>	
	0	Select width from memory via select byte (fixed)
	1	Select width from I/O via select byte (float)

Untuk select byte (Odd Addresses) dapat di tabelkan sebagai berikut :

Tabel 2.6. Tabel *Select Byte (Odd Addresses)*

7	6	5	4	3	2	1	0	
CENTER SELECT				WIDTH SELECT				TYPE=2-7 or WF=0 (FIXED)
	I/O CONT	I/O SELECT CENTER			I/O CONT	I/O SELECT WIDTH		TYPE=2-7 or WF=1 (FLOAT)
ACTION								TYPE=0-1 AF=0 (FIXED)
					I/O CONT	I/O SELECT ACTION		TYPE=0-1 AF=0 (FLOAT)

Kunci :

Width Select	(3:0)	Used address index (E0-EF) for fixed 6-bit. Width value when type=2-7 WF=0
Center Select	(7:4)	Used address index (F0-FF) for fixed 8-bit. Center value when type=2-7 CF=0
I/O SELECT WIDTH	<u>10</u>	
	00	I/O Port 0 as input (type2-7 and WF=1)
	01	I/O Port 1 as input (type2-7 and WF=1)

	10	I/O Port 2 as input (type2-7 and WF=1)
	11	I/O Port 3 as input (type2-7 and WF=1)

I/O Control	<u>2</u>	
	0	Select from input (type 2-7 and WF=1)
	1	Select from output (type 2-7 and WF=1)
I/O Select Center	<u>54</u>	
	00	I/O Port 0 as input (type=2-7 and WF=1)
	01	I/O Port 1 as input (type=2-7 and CF=1)
	10	I/O Port 2 as input (type=2-7 and CF=1)
	11	I/O Port 3 as input (type=2-7 and CF=1)
I/O Control	<u>6</u>	
	0	Select from inputs (type=2-7 and CF=1)
	1	Select from outputs (type=2-7 and CF=1)
ACTION	7-0	8-bit action values to be applied to an output due to winning last term of a rule.
I/O Select Action	<u>10</u>	
	00	I/O Port 0 as input (type=1-0 and AF=1)
	01	I/O Port 1 as input (type=1-0 and AF=1)
	10	I/O Port 2 as input (type=1-0 and AF=1)
	11	I/O Port 3 as input (type=1-0 and AF=1)

I/O Control	<u>2</u>	
	0	Select from inputs (type=1-0 and AF=1)
	1	Select from outputs (type=1-0 and AF=1)

2.2.9. Pewaktuan

Pewaktuan pada AL220 termasuk masukan A/D konverter, prosessor fuzzy dan keluaran multiplex D/A konverter. Kecepatan prosessor merupakan fungsi dari waktu dan nomor clock (1024) diharapkan untuk sampling data lengkap dan siklus prosessing. Maksimum kecepatan *clock* adalah 10 MHz dan minimum adalah 1 MHz.

Waktu Operasi :

1. *Reset*

Ketika pin reset aktif , semua *latch* bersih, keluaran digital *low*, dan keluaran analog menuju kondisi reset. Jika reset aktif untuk 100 clock atau lebih maka sampling masukan analog akan nol , bila kurang maka sampling masukan analog masih mengandung sisa sampel yang dahulu. Ketika reset tidak aktif maka *device* akan mulai melakukan sampling masukan pertama kalinya selama 1024 siklus clock

2. *Konversi Masukan*

Konversi masukan analog ke digital masukan dengan periode selama 256 clock. Total 1024 clock mengkonversi 4 masukan dengan kecepatan sampel maksimal 10 KHz atau 100 μ s.

3. *Waktu Proses*

Ketika Reset tidak aktif maka *device* akan mulai melakukan sampling masukan pertama kalinya selama 1024 *clock cycles*. Konversi masukan analog ke digital dengan periode selama 256 clock. Total 1024 clock mengkonversi 4 masukan dengan kecepatan sampel maksimal 10Khz atau 100 μ s. Pada saat itu *rule* dan variabel fuzzy diproses dengan waktu 12 siklus clock dimana tiap 4 periode .

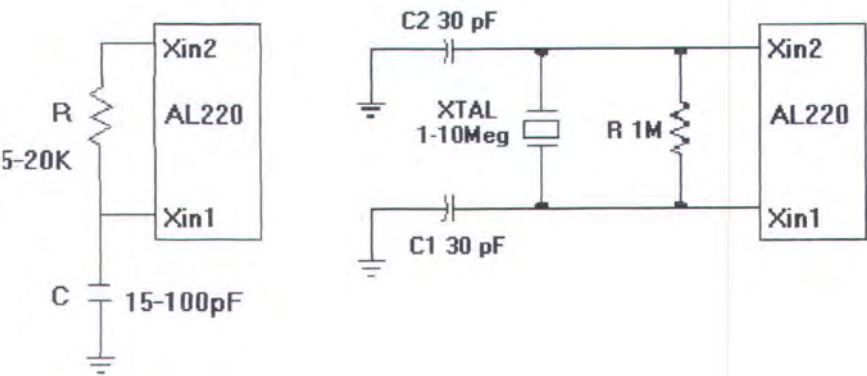
2.2.10. Tundaan *Internal loopback*

Ketika data keluaran di umpan balikkan ke masukan memerlukan waktu inisialisasi siklus sampling selama 1024 clock, kemudian data dari keluaran yang di *loopback* ke masukan di *update* selama proses berlangsung.

2.2.11. Operasi *Prescale*

Prescale aktif untuk *counter* periode waktu dan berbagai keperluan sampling dan proses. Bila dalam posisi *prescale* aktif maka menyebabkan prosessor tidak aktif selama maksimal 1024 *increment clock* .

2.2.12. Rangkaian *Clock* dan *Power Up Reset*



Gambar 2.8. Rangkaian *clock*

Tabel 2.7 Clock Circuit

C1(pF)	R1 = 5K	R1 = 10K	R1 = 18K
15	9.83	6.12	
22	8.05	4.92	
47	4.98	2.85	
100	2.93	1.60	0.945

2.3. Analog to digital converter (ADC)

Modul ADC yang digunakan pada tugas akhir ini adalah menggunakan IC produksi MAXIM MAX 158 , dipilihnya ADC tipe MAX 158 karena hanya memerlukan komponen eksternal yang minimal, mempunyai rangkaian internal multiplexer, track and hold internal dan ADC MAX 158 merupakan ADC CMOS 8 bit dimana mempunyai 8 kanal masukan analog⁷ dengan terseleksi dengan waktu konversi 2.5 μ s dengan internal referensi 2.5 volt dengan error maksimal $\frac{1}{2}$ LSB operasi dengan single +5 volt , tidak menggunakan eksternal clock.

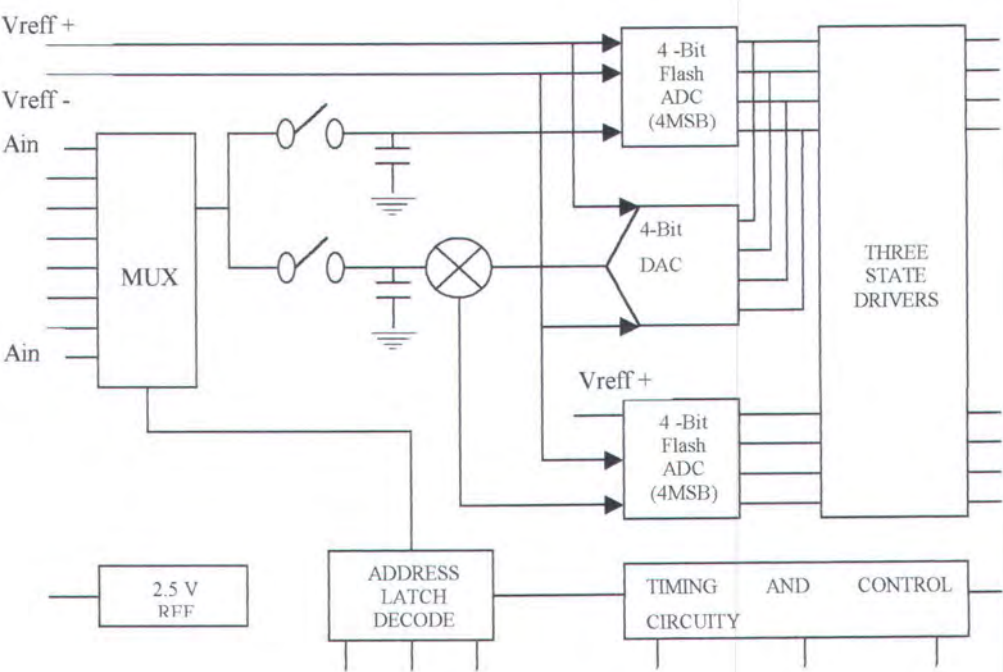
Fungsi masing-masing pin adalah sebagai berikut :

- AIN1-AIN8 : Kanal masukan analog 1 – kanal masukan analog 8
- Ref Out : Referensi keluaran 2.5 volt
- DB0-DB7 : Data keluaran bit 0 – bit 7 bersifat three state
- /RD : Kontrol membaca ADC
- /CS : Kontrol memilih *device*
- /INT : Kontrol interrupt / tanda EOC
- GND : Ground
- RDY : Tanda mulai dan selesai konversi
- A0-A2 : Pemilih kanal masukan data analog
- V_{reff} + : *Upper full scale* keluaran data
- V_{reff} - : *Lower full scale* keluaran data
- VDD : Power supply +5 Volt

⁷ MAX154/MAX158 Data Sheet , Sunnyvale, Maxim Integrated Products, 1999

Tabel 2.8. Tabel kebenaran untuk selektor data masukan A0-A2

A2	A1	A0	KANAL INPUT
0	0	0	AIN 1
0	0	1	AIN 2
0	1	0	AIN 3
0	1	1	AIN 4
1	0	0	AIN 5
1	0	1	AIN 6
1	1	0	AIN 7
1	1	1	AIN 8



Gambar 2.11. Diagram blok ADC MAX 158

Cara operasi ADC MAX 158 dengan memberikan pulsa *low* ke /RD dan /CS selama 600 ns kemudian kita baca /INT apakah sudah selesai konversi (*end of conversion*), bila sudah maka kita baca data valid pada D0-D7 untuk diproses, dan pembacaan ketiga pengkondisi sinyal transducer secara bergantian dengan kanal aktif kanal 1, kanal 2 dan kanal 3 dengan pemberian kode data A2, A1 dan A0.

2.4. Digital to analog converter (DAC)

Digital To Analog Conversion DAC MAXIM MAX 505 merupakan DAC CMOS dengan 4 kanal⁸ keluaran analog terseleksi dengan karakteristik operasi pada power supply +5 volt, keluaran telah terbuffer amplifier, kalibrasi keluaran error 1 LSB, dual buffer input, CMOS/TTL μ P, dan tidak membutuhkan *adjustment* eksternal.

Fungsi masing-masing pin adalah sebagai berikut :

- Vout A – Vout D : DAC keluaran A – DAC keluaran D
- VSS : Power supply negatif
- VDD : Power supply positif
- AGND : Analog ground
- DGND : Digital Ground
- Vref A- Vref D : Tegangan Referensi untuk keluaran DAC A – D
- D0 – D7 : Data DAC 8 bit

⁸ ---, MAX505/MAX506 Data Sheet, Sunnyvale, Maxim Integrated Products, 1999

- /WR : Menulis input data to DAC aktif *low*
- /LDAC : *Latch* input data to DAC aktif *low*
- A0 –A1 : Kanal aktif seleksi

Tabel 2.9. Pengalamatan DAC MAX 505 (*partial list*)

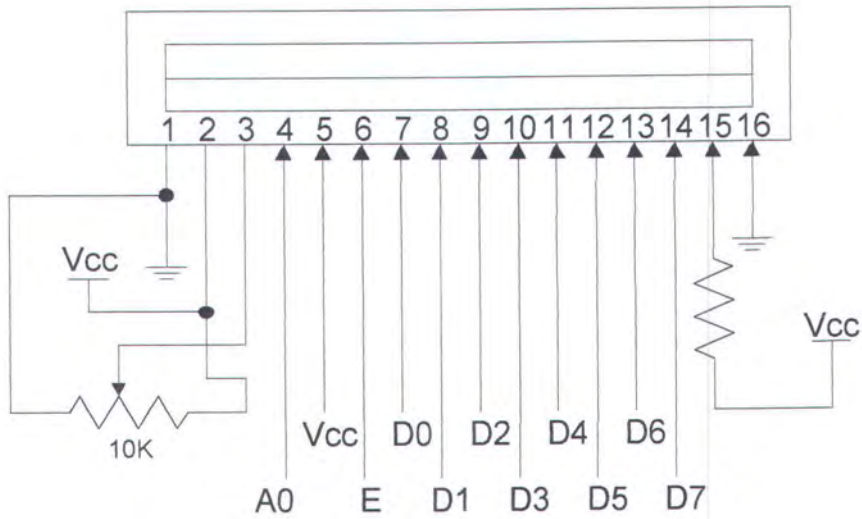
/LDAC	/WR	A1	A0	LATCH STATE
H	H	X	X	Input and DAC data latch
H	L	L	L	DAC A input latch transparant
L	H	X	X	All 4 DAC, DAC latch transparant
L	L	L	L	DAC input A register and lacth transparant
H	L	L	H	DAC B input latch transparant
H	L	H	L	DAC C input latch transparant
H	L	H	H	DAC D input latch transparant

H =High State , L=Low State , X= Don't Care

dimana cara pengoperasiannya adalah dengan memberikan sinyal *low* ke kaki /LDAC dan /WR dengan kanal terseleksi untuk memberikan data secara bergantian dengan melihat tabel kanal aktif A1 dan A0

2.5. Mikrokontroller 80C31BH

Mikrokontroller berfungsi sebagai supervisi untuk proses akses data dengan membaca ADC dan memberi data ke DAC dan di ditampilkan ke LCD. Berikut LCD untuk menampilkan suhu, kelembaban dan tekanan, setting point , *error* masing – masing dan *error* total serta *D error*.



Gambar 2.12. Kontrol Layar LCD

Rangkaian supervisi dapat kita berikan setting point secara eksternal baik suhu, kelembaban, dan tekanan sesuai dengan keinginan kita.



BAB III

PERANCANGAN DAN PEMBUATAN

PERANGKAT KERAS DAN LUNAK

BAB III

PERANCANGAN DAN PEMBUATAN

PERANGKAT KERAS DAN PERANGKAT LUNAK

Dalam membuat sistem tiga dimensi dibutuhkan dua macam perencanaan yang meliputi perancangan perangkat keras dan perencanaan perangkat lunak. Untuk perancangan perangkat lunak akan dibahas pada sub-bab berikutnya.

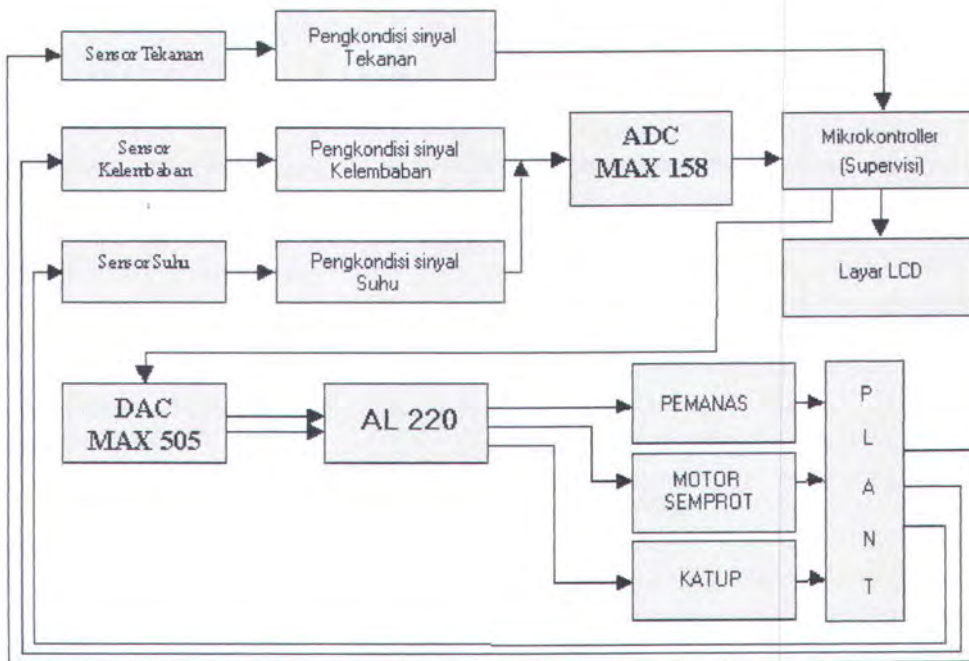
Adapun perancangan perangkat keras adalah :

1. Perancangan keseluruhan sistem.
2. Perancangan dari setiap bagian sistem.

3.1. Perancangan keseluruhan sistem

Adapun keseluruhan sistem dapat dirancang terdiri dari bagian supervisi mikrokontroller 80C31BH , bagian *Fuzzy logic* AL 220 , dan bagian Plant.

Fuzzy yang digunakan adalah tipe AL 220 ,digunakan prinsip *fuzzy logic* tiga dimensi dengan tiga masukan dan empat keluaran , loop tertutup dengan umpan balik *error* dan *D error*.

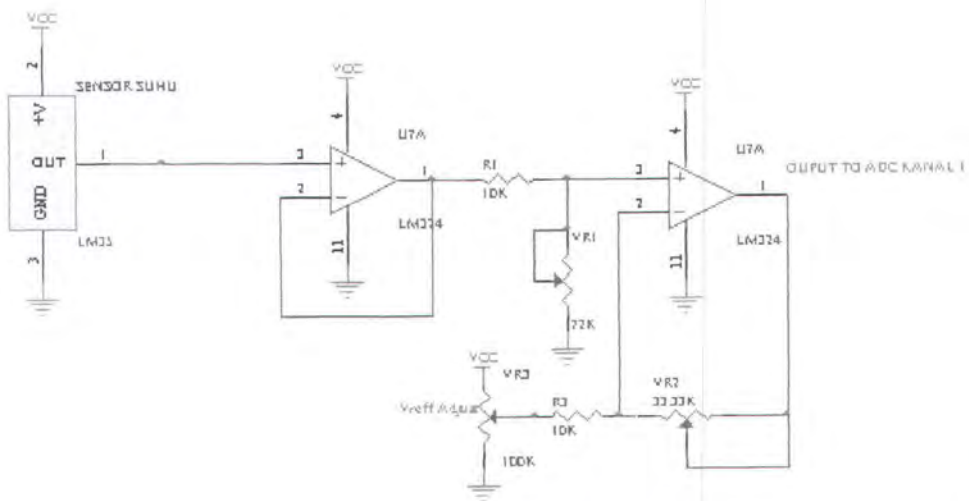


Gambar 3.1. Diagram blok sistem tiga dimensi

3.1.1. Rangkaian sensor suhu dan pengkondisi sinyalnya

Untuk sensor suhu menggunakan LM35 produksi dari National Semikonduktor maka digunakan untuk memperoleh data suhu dengan range temperature dari -55°C sampai 150°C , tegangan keluaran terkalibrasi dalam skala celcius, dengan skala faktor linier $+10.00\text{ mVolt} / ^{\circ}\text{C}$ dengan tegangan operasi dari 4 Volt sampai 30 Volt dengan arus drain kurang dari $60\text{ }\mu\text{ Ampere}$. LM 35 digunakan untuk mengambil data suhu yang dipertahankan pada suhu $+50^{\circ}\text{C}$, sehingga klasifikasi diatas memenuhi standart aplikasinya.

Perancangan pengkondisi sinyal⁹ untuk suhu diperlukan untuk setting pada ADC maksimum adalah + 2,50 Volt, sehingga dikalibrasikan untuk pengkondisi sinyal dari LM35 adalah 1,50 Volt pada suhu +150,00 °C ,dirancang pengkondisi sinyal dari hasil keluaran yang diperoleh di kalibrasi pada mikrokontroller 80C31BH untuk proses selanjutnya.



Gambar 3.2. Pengkondisi sinyal untuk suhu

Dipilih Op-Amp dengan spesifikasi LM 324 sebagai pengkondisi sinyal adalah LM 324 mempunyai karakteristik yang dapat dioperasikan pada tegangan dc +5 V dengan tegangan tunggal saja dan tidak memerlukan offset karena sudah ter-offset nol. Analisa untuk rangkaian Op-Amp pertama yaitu sebagai Voltage Follower adalah :

⁹ Coughlin , Robert F, dan Driscoll, Frederick F. , Penguat Operasional dan Rangkaian Terpadu Linier , Institut Teknologi Wentworth, Diterjemahkan oleh Herman Widodo Soemitro, Jakarta, Penerbit Airlangga ,1992

$$V1 = V_{out} \text{ LM } 35$$

Analisa untuk rangkaian Op-Amp kedua yaitu sebagai Non Inverting Amplifier dengan referensi adalah :

$$V2 = \left(\frac{VR2}{R3} + 1 \right) \times V1$$

Dimana untuk mendapatkan setting pada saat suhu 150 ° C di addjust VR2 sebesar 33.33K dan R3 = 10 K dengan V1=1.5 Volt adalah :

$$V2 = \left(\frac{23.33K}{10} + 1 \right) \times 1.50 = 4.995 \text{ Volt}$$

3.1.2. Rangkaian sensor kelembaban dan pengkondisi sinyalnya

Ada beberapa teori mengenai pengukuran kelembaban¹⁰ yang berhubungan dengan udara diantaranya adalah dengan metode kelembaban relatif yaitu dengan menggunakan *Hygrometer (Psychrometer)* dengan pengukuran terhadap suhu kering dan suhu basah dengan termometer.

¹⁰ John Porges, M.I.Mech.E., M.Inst.F, Handbook of Heating Ventilating and Air Conditioning, Seventh Edition, Newnes-Butterworths, London, Boston Sydney, Wllington, Durban, Toronto. ,p.V.11

Pada tugas akhir ini untuk pengukuran suhu basah dan suhu kering dilakukan dengan menggunakan sensor LM 35¹¹ sebanyak dua buah. LM 35 adalah produksi dari National Semikonduktor maka digunakan untuk memperoleh data suhu dengan range temperature dari $-55\text{ }^{\circ}\text{C}$ sampai $150\text{ }^{\circ}\text{C}$, tegangan keluaran terkalibrasi dalam skala Celcius, dengan skala faktor linier $+10.00\text{ mV} / ^{\circ}\text{C}$ dengan tegangan operasi dari 4 Volt sampai 30 Volt dengan arus current drain kurang dari $60\text{ }\mu\text{A}$. LM 35 digunakan untuk mengambil data suhu yang dipertahankan pada suhu $+50\text{ }^{\circ}\text{C}$, sehingga klasifikasi diatas memenuhi standart typical application.

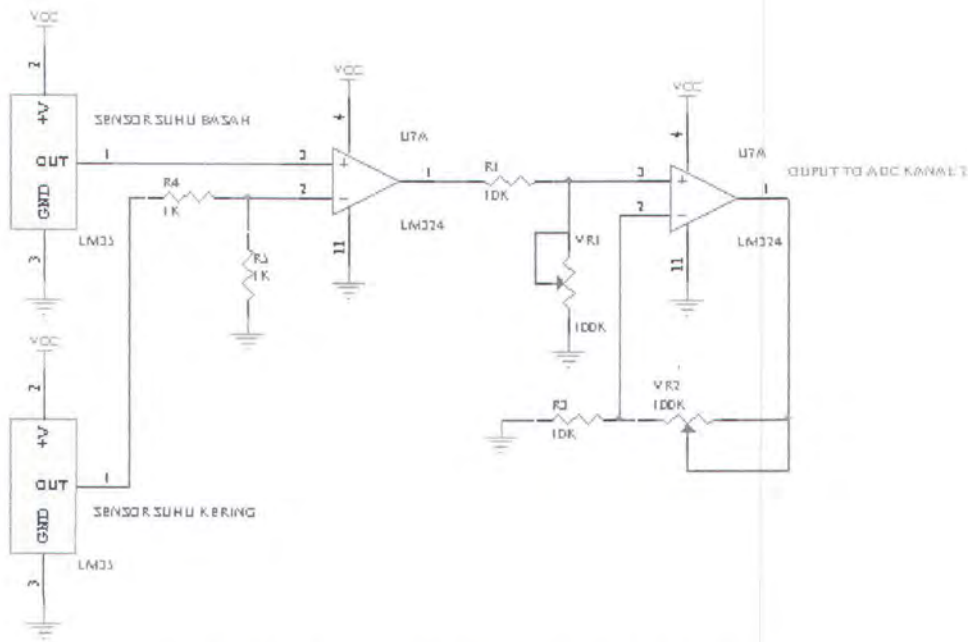
Dipilih Op-Amp dengan spesifikasi LM 324 sebagai DC Processing adalah LM 324 mempunyai karakteristik yang dapat dioperasikan pada tegangan $+5\text{ Vdc}$ dengan tegangan tunggal saja dan tidak memerlukan offset karena sudah ter-offset internal.

Proses pemasangan sensor adalah dengan memasang LM 35 yang pertama pada suhu basah, dan LM 35 yang kedua pada suhu kering didekatnya, besar kelembaban diperoleh dari selisih keduanya dengan memperhitungkan suhu selisihnya diperoleh data kelembaban.

Misal diperoleh data bahwa pada suhu basah uap air $52\text{ }^{\circ}\text{C}$, sedangkan pada suhu kering $50\text{ }^{\circ}\text{C}$ maka diperoleh data kelembaban relatif sebesar 87 % dari tabel. Sehingga dirancang DC Processing untuk kelembaban dengan setting untuk

¹¹ National Semiconductor, 1995 National Data Acquisition Databook, USA, National Semiconductor, 1995, p.5-12

memperoleh data kelembaban yang persisi dan sesuai tabel *relative humidity wet and dry bulb temperature*¹².



Gambar 3.3. Pengkondisi sinyal untuk kelembaban

Pada rangkaian diatas untuk Op-Amp pertama berfungsi sebagai differensial, sedangkan Op-Amp kedua berfungsi sebagai Non Inverting Amplifier, dengan perhitungan misal untuk suhu kering 50 °C dan suhu basah 52 °C maka :

¹² John Porges, M.I.Mech.E., M.Inst.F, Handbook of Heating Ventilating and Air Conditioning, Seventh Edition, Newnes-Butterworths, London, Boston Sydney, Wllington, Durban, Toronto

$$V_{out} = V_1 - V_2$$

$$V_{out} = (0.52) - \left(\frac{1K}{1K + 1K} \right) (0.50)$$

$$V_{out} = (0.52) - (0.25) = 0.27$$

Dari data tabel diperoleh kelembaban relatif 87% , untuk rangkaian Non Inverting Amplifier di adjust harga $VR_2 = 22 K$ maka

$$V_{out\ 2} = \left(\frac{VR_2}{R_3} + 1 \right) \times V_{out\ 1}$$

$$V_{out\ 2} = \left(\frac{22 K}{10 K} + 1 \right) \times 0.27$$

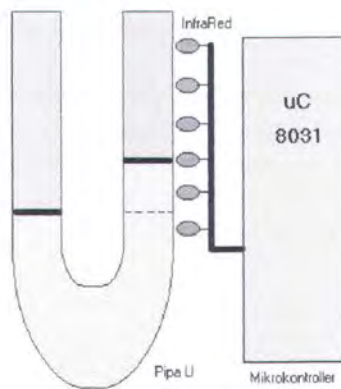
$$V_{out\ 2} = 0.864 \text{ Volt}$$

Dari hasil keluaran yang diperoleh di kalibrasi pada mikrokontroller 80C31BH untuk proses selanjutnya.

3.1.3. Rangkaian sensor tekanan dan pengkondisi sinyalnya

Pengukuran tekanan dapat dilakukan dengan mengkalibrasi sensor tekanan dengan hukum fisika pipa u manometer gas ¹³ yang ada dengan pengkondisi sinyalnya sebagai berikut :

¹³ Team MGMP IPA KOTAMADIA Surabaya, Pelajaran Fisika (Kurikulum SMP 1994) kelas 1. , p.35



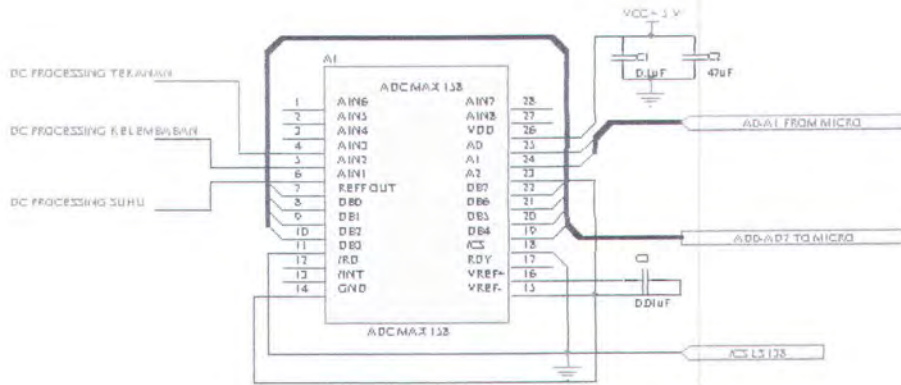
Gambar 3.4. Pengkondisi sinyal untuk tekanan

3.1.4. Rangkaian *analog to digital converter*

Perancangan rangkaian *analog to digital converter* (ADC) dengan menggunakan tipe IC MAX 158 adalah :

1. Memperhatikan V_{ref+} dan V_{ref-} sebagai definisi nol dan full scale ADC
2. Pemasangan kapasitor $0.01 \mu F$ sebagai kapasitor bypass ke ground untuk menghilangkan high frequency keluaran impedance.
3. Pemasangan kapasitor elektrolit $47 \mu F$ dan kapasitor keramik $0.1 \mu F$ untuk kompensasi error konversi dan kondisi tidak stabil.

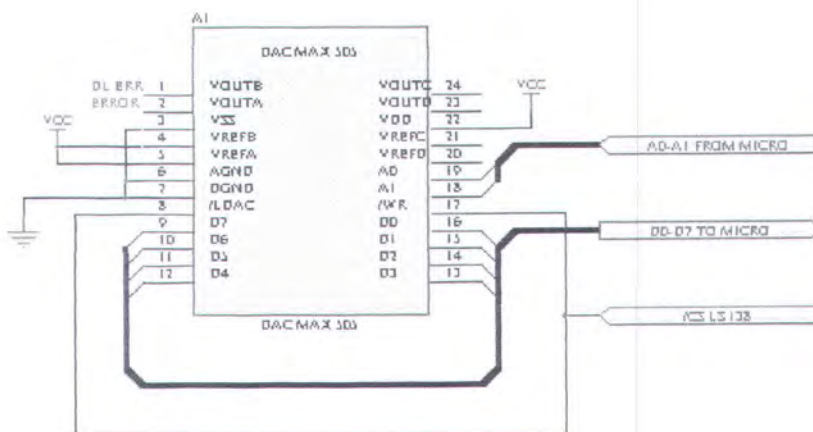
Sehingga di rancang ADC MAX 158 dengan tipe internal referensi sebagai berikut :



Gambar 3.5. Rangkaian ADC MAX 158

3.1.5. Rangkaian *digital to analog converter*

Perancangan dari rangkaian *digital to analog converter* (DAC) dengan menggunakan tipe IC MAX 505 dengan standart rangkaian sebagai berikut :



Gambar 3.6. Rangkaian DAC MAX 505

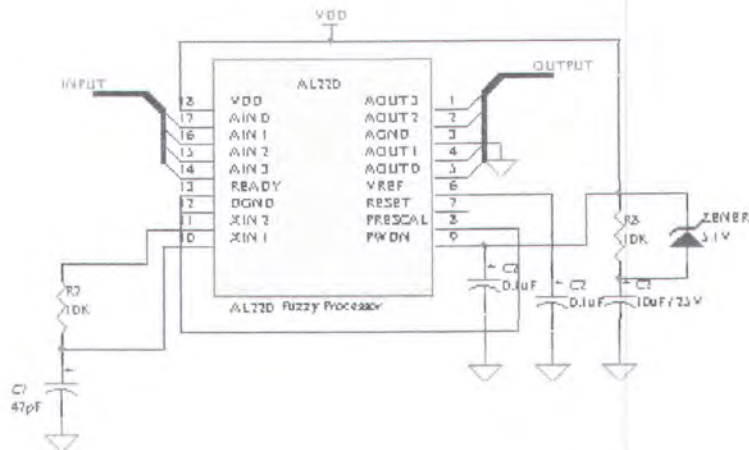
3.1.6. Rangkaian mikrokontroller 80C31BH

Rangkaian mikrokontroller dengan tipe IC 80C31BH¹⁴ sebagai kontroller untuk bagian supervisi sebagai pemroses data dari sinyal yang diambil dari masing-masing sinyal pengkondisi melalui ADC, kemudian diproses untuk menghasilkan sinyal error dan sinyal delay error yang diubah dari data digital ke analog melalui DAC untuk di umpankan ke IC *Fuzzy logic* tipe AL220.

¹⁴ ---, Embeded Controller Hanbook data sheet , 1988 , intel

3.2. Rangkaian *fuzzy logic* AL220

Rangkaian *fuzzy logic* AL220 dengan perancangan sebagai berikut :



Gambar 3.7. Rangkaian AL 220

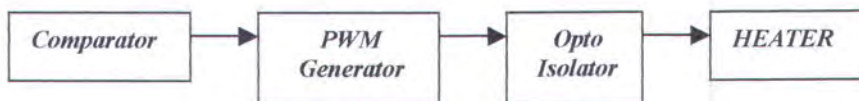
Perancangan AL220 diatas dapat kita gunakan untuk aplikasi *fuzzy logic* tiga dimensi dengan memanfaatkan kanal masukan Ain0 sampai Ain3 dengan kanal keluaran Aout0 sampai Aout3.

3.3. Rangkaian Aktuator

Rangkaian aktuator dirancang untuk menggerakkan suatu alat dimana karena terjadinya perubahan pada alat akan mengakibatkan terjadinya perubahan pada plant. Dalam tugas akhir ini dirancang sebanyak tiga macam aktuator yaitu masing-masing untuk aktoator x , y , dan z dimana untuk aktuator Suhu, Kelembaban dan Tekanan.

3.3.1 Rangkaian aktuator untuk plant suhu

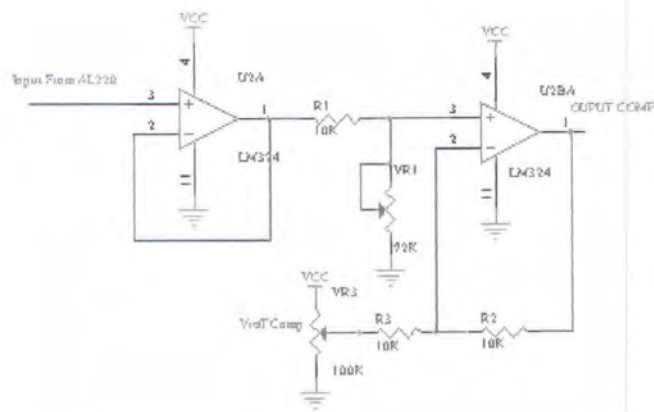
Rangkaian aktuator suhu berfungsi untuk menjaga kondisi panas heater tetap konstan pada suhu setting dengan cara mensampling tegangan masukan kontrol heater dengan sinyal keluaran dari *fuzzy logic* AL220. Maka diperlukan suatu rangkaian aktuator yang terdiri dari komparator, pembuat sinyal *PWM*, dan rangkaian Opto triac untuk pengemudi triac.



Gambar 3.8. Diagram blok aktuator untuk plant suhu

3.3.1.1 Rangkaian komparator

Rangkaian ini berfungsi untuk melakukan komparator terhadap sinyal keluaran dari AL 220 sehingga pada setting Xout dapat bekerja sesuai dengan yang diharapkan.

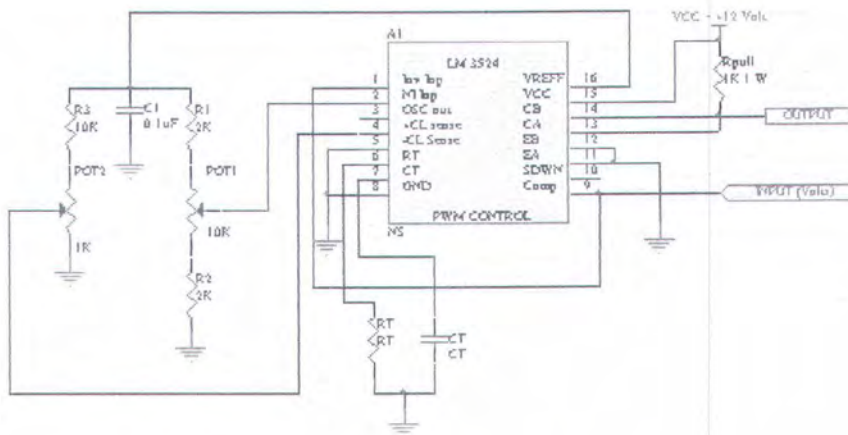


Gambar 3.9. Rangkaian komparator

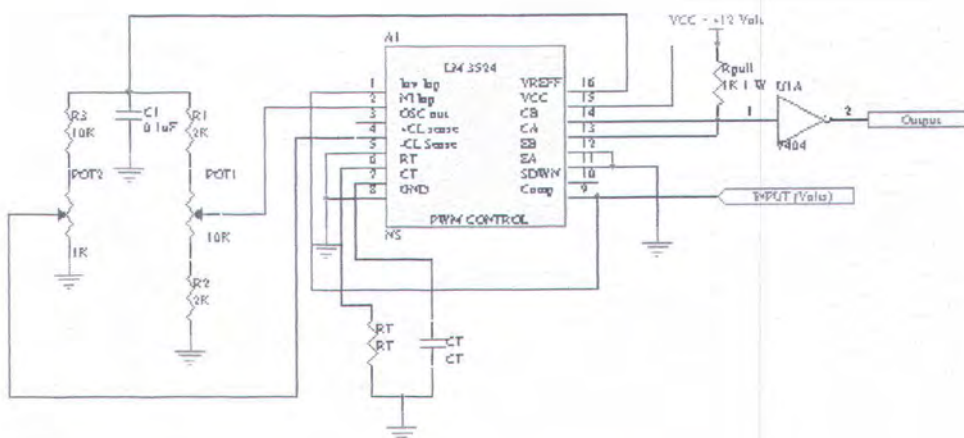
3.3.1.2 Rangkaian *Pulse Width Modulation*

Rangkaian ini berfungsi untuk mengubah tegangan dc volt ke dalam bentuk *Pulse Width Modulation* ¹⁵ dimana dengan naiknya tegangan DC Volt maka akan menghasilkan keluaran dengan lebar pulsa yang lebih besar, dimana lebar pulsa bergerak dari *duty cycle* $\approx 0\%$ sampai *duty cycle* $\approx 99\%$ dengan amplitudo tegangan sebesar 12 Volt.

¹⁵ J. Michael Jacob , Industrial Control Electronic Application Design , Prentice Hall International , Inc

Gambar 3.10. Rangkaian *PWM*

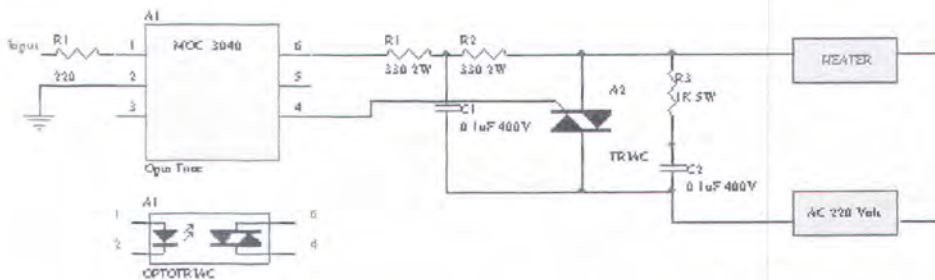
Keluaran dari rangkaian *PWM* diumpankan ke logika TTL 74LS04 yaitu *Inverter* dengan membagi tegangan masukan agar di peroleh tegangan keluaran sebesar 5 Volt, agar bisa di umpankan ke Opto Triac dengan range masukan 0 – 5 volt.

Gambar 3.11. *PWM* dengan keluaran *inverting*

3.3.1.3 Rangkaian Opto Triac

Rangkaian opto triac berfungsi untuk mengontrol dengan memberikan sampling terhadap tegangan AC ≈ 220 Volt sehingga diperlukan rangkaian isolasi yang baik antara rangkaian kontrol dengan rangkaian penggerak.

Untuk kontrol heater digunakan triac sebagai saklar AC, untuk memicu gate pada triac diperlukan rangkaian yang mampu memberi arus dan tegangan yang cukup pada gate, tanpa melampaui dissipasi daya maksimum pada gate. Selain itu karena triac terhubung jaringan tegangan tinggi maka rangkaian harus mampu untuk melindungi rangkaian kontrol.



Gambar 3.12. Rangkaian Opto triac

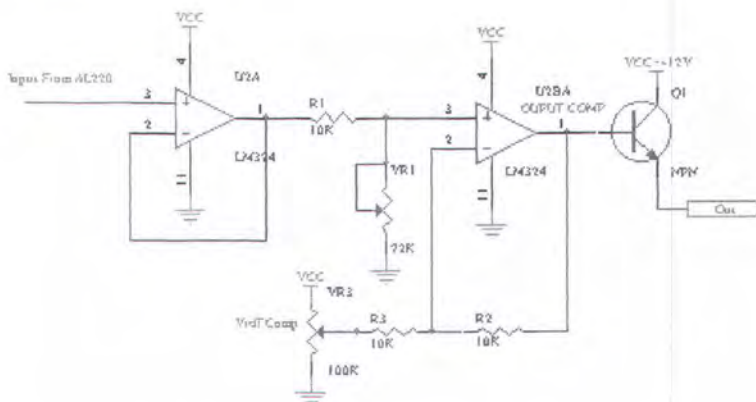
Untuk mengisolasi rangkaian kontrol dari jaringan listrik maka dipakai sebuah isolator optik. Isolator optik merupakan isolator yang mampu mengemudikan triac dengan memenuhi syarat sebagai isolasi, bentuknya kecil, murah, dan mudah dinyalakan dengan komparator atau logika digital. Isolator optik didalamnya terdapat saklar bilateral silikon yang diaktifkan cahaya, saklar ini

dikopel secara optik oleh internal LED, jika LED ini menyala maka saklar akan diaktifkan.

Digunakan MOC3041, yang merupakan sebuah opto isolator dengan deteksi zero crossing detector internal dan tegangan isolasi maksimum 7500 volt, dapat digunakan untuk mensaklar triac, LED harus menyala, dan jaringan listrik sedang tinggi maka arus gate yang dialirkan hanya sebesar $20\ \mu\text{A}$.

3.3.2 Rangkaian aktuator untuk plant kelembaban

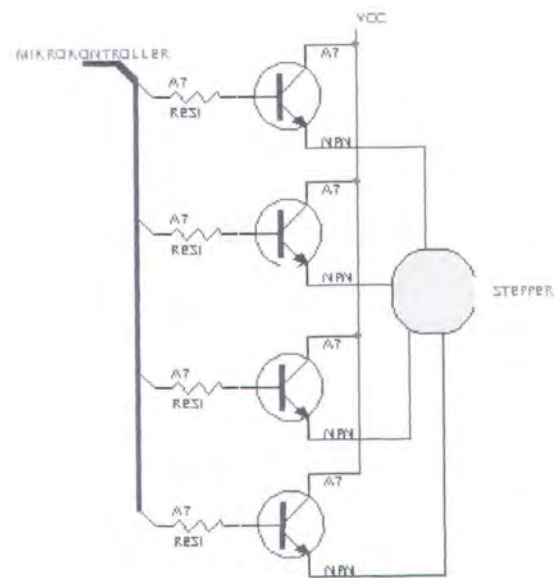
Rangkaian aktuator kelembaban berfungsi untuk menjaga kondisi kelembaban ruangan tetap konstan pada lembab setting dengan cara memberikan kontrol pada motor semprot dengan sinyal keluaran Yout dari *fuzzy logic* AL220. Maka diperlukan suatu rangkaian aktuator yang terdiri dari komparator dan rangkaian penggerak motor semprot



Gambar 3.13. Rangkaian aktuator untuk plant Kelembaban

3.3.3 Rangkaian aktuator untuk plant tekanan

Rangkaian aktuator tekanan berfungsi untuk menjaga kondisi tekanan ruangan tetap konstan pada tekanan setting dengan cara memberikan kontrol pada motor katup dengan sinyal keluaran Zout dari *fuzzy logic* AL220 . Maka diperlukan suatu rangkaian aktuator pada mikrokontroller untuk rangkaian penggerak motor stepper.



Gambar 3.14. Rangkaian aktuator untuk plant tekanan

3.4 Perancangan perangkat lunak *fuzzy logic* tiga dimensi

3.4.1 Teori suhu, tekanan, dan kelembaban

Teori yang mendasari adanya pendekatan penyelesaian masalah dengan menggunakan metode *fuzzy logic* diperlukan adanya pendekatan dengan metode *heuristic* yaitu metode yang mendekati persamaan matematika dan atau fisika dengan metode *fuzzy logic*. Pada Tugas akhir ini di tinjau beberapa metode mengenai suhu, tekanan dan kelembaban untuk dicapai suatu pendekatan dalam bentuk pemilihan model matematika untuk dibawa ke model *rule-rule fuzzy logic*.

- Hukum yang menyatakan *kesetaraan* antara *Volume, Tekanan, dan Suhu* dengan meninjau pada *Normal Temperature dan Pressure (N.T.P)* adalah untuk memudahkan maka kita tinjau dengan standart pada suhu dan tekanan yang di anggap pada kondisi normal temperature dan pressure.

Normal Temperature = 0 °C = 32 °F

Normal Pressure = 14.7 lb.per sq.in = 1 atm = $1.013 \times 10^5 \text{ N/m}^2$

Dimana : W = Berat dari gas

V1 V2 = Inisial and respon volume akhir

P1 P2 = Inisial and respon volume akhir

T1 T2 = Inisial and suhu mutlak akhir

R = Konstanta gas

Rm = Konstanta campuran gas

- Hukum yang berlaku ada beberapa analisa¹⁶ yaitu :

1. Hukum Boyle

Dengan suhu konstan maka berlaku hukum

$$P_1.V_1 = P_2.V_2$$

$$P.V = \text{Konstan}$$

$$\frac{P_1}{P_2} = \frac{V_2}{V_1}$$

2. Hukum Gay-Lussac's

Dengan volume konstan maka berlaku hukum

$$\frac{P_1}{P_2} = \frac{T_1}{T_2}$$

3. Hukum Charles

Dengan tekanan konstan maka berlaku hukum :

$$\frac{V_1}{V_2} = \frac{T_1}{T_2}$$

4. Hukum kombinasi Boyle, Charles dan Gay-Lussac :

$$\frac{P_1.V_1}{T_1} = \frac{P_2.V_2}{T_2}$$

¹⁶ John Porges, M.I.Mech.E., M.Inst.F, Handbook of Heating Ventilating and Air Conditioning, Seventh Edition , Newnes-Butterworths ,London ,Boston Sydney , Wllington, Durban, Toronto. , p.iv.5

5. Hukum Gas Umum

Pada gas mulia berlaku hukum :

$$P.V = w.R.T$$

dimana : $wR = 1.985 \text{ B.t.u} / \text{lb/mol } ^\circ\text{F}$ (pada N.T.P 1kg mol)

$$= 1.985 \text{ Kcal} / \text{kg.mol } ^\circ\text{C}$$

$$= 8.3143 \text{ kJ} / \text{kg.mol } ^\circ\text{C}$$

- Hukum yang menyatakan hubungan antara suhu tekanan dan kelembaban dari hubungan diatas dapat ditemukan hubungan :

1. *Dry Bulb Temperature*
2. *Wet Bulb Temperature*
3. *Dew Point*
4. *Relative Humidity*
5. *Specific Humidity*
6. *Vapour Pressure*
7. *Total Heat*

Penjelasan hubungan tersebut adalah :

1. *Dry Bulb Temperature* adalah suhu di udara yang ditunjukkan oleh termometer yang tidak dipengaruhi oleh kelembaban udara.
2. *Wet Bulb Temperature* adalah suhu adiabatic saturasi, yang ditunjukkan suhu terendah oleh termometer kelembaban.

3. *Dew Point Temperature* adalah suhu yang ditunjukkan oleh udara dengan kelembaban dan titik kondensasi uap air.

- Dengan hukum gas umum diperoleh hubungan :

1. Densitas dari udara kering :

$$d_a = \frac{1.323.P_a}{T} (\text{lb.per.cu.ft}) = \frac{0.470.P_a}{T} (\text{kg.per.m}^3)$$

2. Densitas Uap Air

$$d_w = \frac{0.821.P_a}{T} (\text{lb.per.cu.ft}) = \frac{0.293.P_a}{T} (\text{kg.per.m}^3)$$

3. Densitas Campuran (1) dan (2)

$$d = d_a - d_w$$

- *Humidity* adalah bentuk dari persentase quantity uap air di udara
- *Absolute Humidity* adalah berat aktual dari uap air dalam satuan grain atau lb per cu.ft. dari campuran atau gram per kilogram
- *Specific Humidity* adalah berat aktual dari uap air dalam gr atau lb per 1 lb dari udara kering atau gram per 1 kg.

$$X = 4354. \frac{\phi P_{ws}}{P - P_{ws}} \text{ grain/lb} = 622 \frac{\phi P_{ws}}{P - P_{ws}} \text{ grammes/kg}$$

- *Humidity* spesifik dari Udara Jenuh adalah

$$X_s = 4354. \frac{P_{ws}}{P - P_{ws}} \text{ grain/lb} = 622 \frac{P_{ws}}{P - P_{ws}} \text{ grammes/kg}$$

- Kelembaban relatif adalah
 1. Suatu ratio dari bagian aktual dari tekanan dari uap air (suhu basah dipengaruhi uap air) terhadap tekanan uap air pada saat tekanan pada udara saturasi saat dry bulb temperature (suhu kering yang tidak dipengaruhi uap air).
 2. Suatu ratio dari densitas uap terhadap densitas uap pada udara saturasi saat dry bulb temperature.
 3. Suatu ratio dari uap lembab aktual pada udara pada volume terhadap uap lembab dibutuhkan untuk volume saturasi
- Sehingga diekspresikan dalam persamaan matematika :

$$\phi = \frac{P_{WA}}{P_{WS}} = \frac{d_W}{d_{WS}} = \frac{X}{X_S} (\%)$$

- Udara Saturasi adalah menunjukkan jumlah maksimum dari air pada kondisi temperatur . Beberapa suhu udara rendah akan menyebabkan kondensasi dari uap air.

dimana :

V = Volume udara kering -campuran uap (cu.ft. atau m³)

W = Berat udara-campuran uap (lb or kg)

P₂ = Tekanan parsial dari uap

P_{wa} = Tekanan aktual parsial dari uap air

P_{ws} = Tekanan saturasi dari uap air

P = Total tekanan

T = Dry bulb temperature

$T = t + 460$ = Suhu mutlak °F ($t+273^{\circ}\text{C}$)

ϕ = Kelembaban relatif (%)

X = Aktual kelembaban spesifik (gr / lb atau g /kg)

X_s = Saturasi kelembaban spesifik (gr / lb atau g /kg)

d_a = Densitas udara kering (lb / cu.ft atau kg / m³)

d_w = Densitas uap air (lb / cu.ft atau kg / m³)

d = Densitas udara dan campuran uap air (lb / cu.ft atau kg / m³)

R = Gas konstan pada udara kering $R = 53.34$ pada uap air $R = 86.00$

1mm dari Mercury = $1.33 \times 10^2 \text{ N/m}^2$

Metode *heuristic* dapat digunakan untuk penyelesaian masalah diatas dan merupakan salah satu metode yang digunakan untuk mendekati logika matematika dan atau fisika ke dalam logika *fuzzy logic*.

Untuk suhu dan Tekanan :

Hukum *Gay lussac* (volume dianggap konstan)

$$\frac{P_1}{P_2} = \frac{T_1}{T_2}$$

Pada suhu awal $T_1 = 30^{\circ}\text{C}$ dengan perubahan untuk suhu plant $T_2 = 50^{\circ}\text{C}$ maka dapat dibentuk persamaan dengan melihat perubahan pada P_2 (tekanan untuk plant)

$$\frac{P_1}{P_2} = \frac{T_1}{T_2}$$

$$\frac{P_1}{P_2} = \frac{30}{50}$$

$$P_1 = 0.6 \times P_2$$

Dengan kenaikan suhu maka tekanan juga semakin bertambah dengan ratio yang linier.

Untuk Suhu dan Kelembaban :

Adanya kelembaban relatif dalam hubungan

$$\phi = \frac{P_{WA}}{P_{WS}} = \frac{d_W}{d_{WS}} = \frac{X}{X_S} (\%)$$

Dimana perubahan tekanan linier terhadap perubahan suhu (Hukum *Gay Lussac*) dengan menganggap volume konstan, maka

$$\phi = \frac{d_W}{d_{WS}} (\%)$$

$$\phi = \frac{0.293 \times P_W}{0.470 \times P_A} (\%)$$

$$\phi = 0.623 \times \frac{P_W}{P_A} (\%)$$

$$\phi = 0.623 \times \frac{P_2}{P_1} (\%)$$

$$\phi = 0.623 \times \frac{T_2}{T_1} (\%)$$

Perubahan suhu basah terhadap suhu kering mempengaruhi perubahan kelembaban.

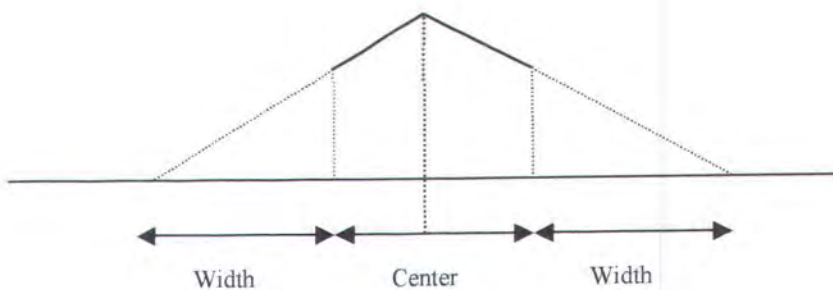
3.4.2 *Heuristic* perangkat lunak *fuzzy* tiga dimensi

Perancangan perangkat lunak dari *fuzzy logic* dapat dilakukan dengan logika *fuzzy* tiga dimensi.

3.4.2.1 Perancangan fungsi keanggotaan untuk Suhu

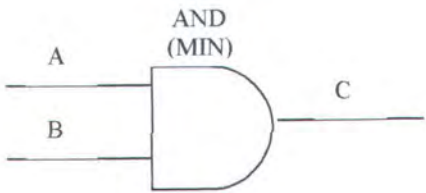
Perancangan fungsi keanggotaan untuk pengaturan suhu (dimensi x) secara loop tertutup dengan memperhatikan faktor masukan *error* dan *D error* yang mana *D error* adalah *last error* dari DAC MAX 505.

Rule-rule pada logika *fuzzy* dapat di buat sesuai dengan keinginan kita dimana suhu yang akan kita kontrol dapat kita setting untuk menentukan besarnya suhu rata-rata yang diharapkan, apabila kita ingin membuat aksi kontrol pada Plant dengan suhu pada 50 °Celcius maka kita setting *rule – rule* dengan fungsi keanggotaan untuk mengolah data masukan untuk menghasilkan data keluaran yang sesuai yaitu pada setting suhu 50°C, dan besarnya osilasi *error* dapat kita setting dengan memberikan toleransi *error* diantara 50 °C dengan mengatur lebar pada fungsi keanggotaan.



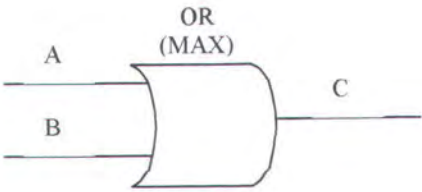
Gambar 3.15. Variabel fungsi keanggotaan

Kerja dari *fuzzy logic* adalah mengikuti dua teorema evaluasi *rule* yaitu *MAX* and *MIN* yang di jelaskan sebagai berikut :



Tabel 3.1. Tabel teori *MIN*

A	B	C
0.1	0.6	0.1
0.8	0.3	0.3
0	0	0
0	1	0
1	0	0
1	1	1



Tabel 3.2. Tabel teori *MAX*

A	B	C
0.1	0.6	0.6
0.8	0.3	0.8
0	0	0
0	1	1
1	0	1
1	1	1

NOT
(COMPLEMENT)



Tabel 3.3 Tabel teori *NOT*

A	C
0.3	0.7
0.9	0.1
0	1
1	0
0.5	0.5

Sehingga dapat dirancang variabel fuzzy pada FAE Software untuk suhu sebagai berikut :

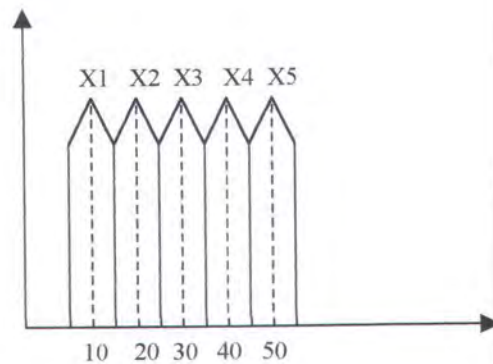
State is X1 (10 , 0 , 1 , SI)

State is X2 (20 , 0 , 1 , SI)

State is X3 (30 , 0 , 1 , SI)

State is X4 (40 , 0 , 1 , SI)

State is X5 (50 , 0 , 1 , SI)



Gambar 3.16. Fungsi keanggotaan dimensi x untuk suhu

Penjelasan fungsi keanggotaan diatas adalah sebagai berikut :

1. State berawal pada state zero, dimana sistem yang dirancang bersifat loop tertutup , dari x ke y ke z kemudian *loopback*. Pada state Zero ini sistem akan *idle* apabila sistem mendeteksi masukan *error* dan *D error* sama dengan nol.

2. State X1 distate ini akan di bandingkan apakah *error* lebih besar dari nol atau lebih kecil sama dengan nol, apabila sistem lebih kecil sama dengan nol maka sistem akan *idle* , berarti tidak ada *error* atau kondisi setting telah terpenuhi.
3. State X2 akan menambah Xout untuk di bandingkan bila *error* lebih besar nol dengan *delayed error*, bila penambahan tersebut *error* lebih kecil sama dengan *delayed error* maka di koreksi apakah *error* lebih besar dari nol.
4. Bila *error* masih besar maka kembali ke state X2, bila *error* lebih kecil sama dengan *D error* maka mengurangi Xout di state X4 kemudian di koreksi apakah *error* lebih besar dibandingkan dengan *delayed error*, bila ya maka kembali ke state X2, bila lebih kecil sama dengan *D error* maka berarti aksi yang dilakukan benar dan kembali ke state X4.
5. State X5 mengkoreksi apakah *error* terlalu besar, bila terlalu besar dan batas iterasi maksimum maka state koreksi pada state Y1 dimana proses pada State Y dan State Z sama dengan pada State X

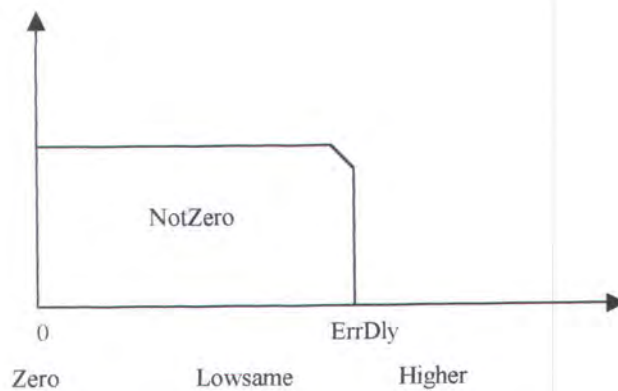
Penjelasan fungsi keanggotaan untuk *error* dan *delayed error* :

Prinsip dari Rule-rule penyusun fungsi keanggotaan untuk masukan *error* dan *D error* adalah dengan fungsi keanggotaan *floating* sehingga mempunyai beberapa keuntungan untuk penggunaan metode fungsi keanggotaan *floating* adalah

1. Menghemat penggunaan memori pada *fuzzy logic* AL220.

2. Dapat membandingkan atau mencari perbedaan dua buah perbedaan masukan pada logika fuzzy, apakah lebih besar, lebih kecil atau sama.
3. Dapat mencari kedekatan dengan pusat yang dapat bertambah atau berkurang sesuai dengan masukan *floating*.
4. Dapat mengubah respon sensor tiap saat dengan perbandingan terhadap setting tegangan.
5. Mengkalibrasi *rule-rule* dengan cek pada derajat koreksi masukan.
6. Dapat perbandingan mencari kesamaan atau kedekatan *error* masukan lebih cepat, dengan akselesari keluaran lebih cepat.

Perancangan fungsi keanggotaan untuk *error* adalah :



Gambar 3.17. Fungsi keanggotaan untuk *error*

Penjelasan fungsi keanggotaan diatas adalah :

1. “ **Error is Zero** “ adalah mengakibatkan aksi kontrol apabila error nol maka mengerjakan aksi state idle
2. “ **Error is NotZero** “ adalah mengakibatkan aksi untuk perbandingan ada error .

3. “ **Error is Lowsame** “ adalah mengakibatkan adanya aksi kontrol pembanding error lebih kecil sama dengan error delay
4. “ **Error is Higher** “ adalah mengakibatkan adanya aksi kontrol pembanding error lebih besar dengan error delay

3.4.2.2. Perancangan fungsi keanggotaan untuk kelembaban

Perancangan fungsi keanggotaan untuk pengaturan kelembaban (dimensi Y) secara adaptif loop tertutup adalah dengan memperhatikan faktor masukan *error* dan *D error* yang dipengaruhi oleh state dari proses sebelumnya yaitu dimensi X.

Sehingga dapat dirancang variabel fuzzy pada FAE Software untuk suhu sebagai berikut :

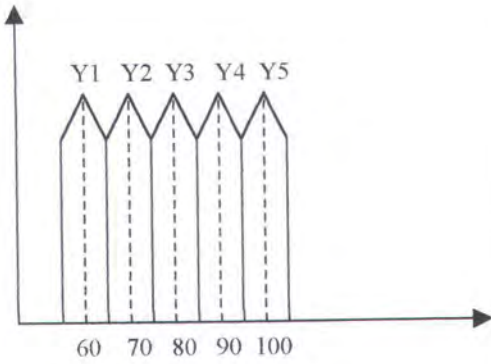
State is Y1 (60 , 0 , 1 , SI)

State is Y2 (70 , 0 , 1 , SI)

State is Y3 (80 , 0 , 1 , SI)

State is Y4 (90 , 0 , 1 , SI)

State is Y5 (100 , 0 , 1 , SI)



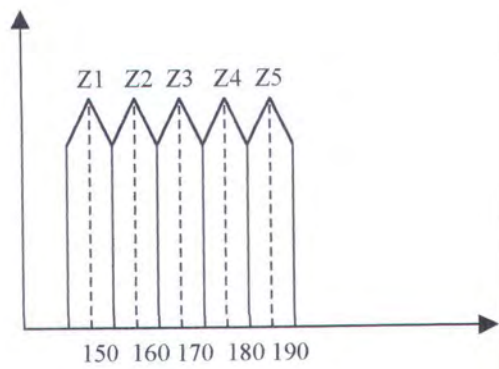
Gambar 3.18. Fungsi keanggotaan dimensi Y untuk kelembaban

3.4.2.3. Perancangan fungsi keanggotaan untuk tekanan

Perancangan fungsi keanggotaan untuk pengaturan tekanan (dimensi Z) secara adaptif loop tertutup adalah dengan memperhatikan faktor masukan *error* dan *D error* yang dipengaruhi oleh state dari proses sebelumnya yaitu dimensi Y.

Sehingga dapat dirancang variabel fuzzy pada FAE Software untuk suhu sebagai berikut :

- State is Z1 (110 , 0 , 1 , SI)
- State is Z2 (120 , 0 , 1 , SI)
- State is Z3 (130 , 0 , 1 , SI)
- State is Z4 (140 , 0 , 1 , SI)
- State is Z5 (150 , 0 , 1 , SI)



Gambar 3.19. Fungsi keanggotaan dimensi Z untuk tekanan

3.4.2.4. Penjelasan rule-rule perangkat lunak *Fuzzy logic*

Rule no 1

If Setpoint is Off Then Xout = 0#

Setting sistem idle

Rule no 2

If Setpoint is Off Then Yout = 0#

Setting sistem idle

Rule no 3

If Setpoint is Off Then Zout = 0#

Setting sistem idle

Rule no 4

If Setpoint is On and State is Zero Then State = 10#

Jika sistem tidak idle dan state masih nol maka state diupdate menjadi State X1

Rule no 5

If Error is Zero and State is X1 Then State = 0

Jika error telah nol maka sistem idle

Rule no 6

If State is X1 Then State = 20:

Jika State X1 maka action last rule ke State X2

Rule no 7

If State is X2 Then Xout + 1#

Update Xout dengan accumulate pada State X2

Rule no 8

If State is X2 Then State = 30#

Update State X2 menjadi State X3

Rule no 9

If Error is LowSame and State is X3 Then State = 40

Jika Error lebih kecil sama dengan Last Error atau *D error* pada state X3 maka State sekarang X4

Rule no 10

If Error is Higher and State is X3 Then State = 20

Jika Error lebih besar dari Last Error atau *D error* pada state X3 maka State sekarang X2

Rule no 11

If State is X4 Then State = 50:

Jika State X4 maka action last rule State X5

Rule no 12

If State is X5 Then Xout - 1#

State X5 mengupdate Xout dengan accumulate

Rule no 13

If Error is LowSame and State is X5 Then State = 40

Jika Error lebih kecil sama dengan Last Error atau *D error* pada state X5 maka State sekarang X4

Rule no 14

If Error is Higher and State is X5 Then State = 60:

Jika Error lebih besar dari Last Error atau *D error* pada state X5 maka State sekarang Y1

Rule no 15

If Xout is Higher Then Xout = 30#

Jika Xout pada setting point atau level iterasi maksimum maka update Xout = 30

Rule no 16

If Error is Zero and State is Y1 Then State = 0

Rule no 17

If State is Y1 Then State = 70:

Rule no 18

If State is Y2 Then Yout + 40#

Rule no 19

If State is Y2 Then State = 80#

Rule no 20

If Error is LowSame and State is Y3 Then State = 90

Rule no 21

If Error is Higher and State is Y3 Then State = 70

Rule no 22

If State is Y4 Then State = 100:

Rule no 23

If State is Y5 Then Yout - 20#

Rule no 24

If Error is LowSame and State is Y5 Then State = 90

Rule no 25

If Error is Higher and State is Y5 Then State = 110:

Rule no 26

If Yout is Higher Then Yout = 30#

Rule no 27

If Error is Zero and State is Z1 Then State = 0

Rule no 28

If State is Z1 Then State = 120:

Rule no 29

If State is Z2 Then Zout + 20#

Rule no 30

If State is Z2 Then State = 130#

Rule no 31

If Error is LowSame and State is Z3 Then State = 140

Rule no 32

If Error is Higher and State is Z3 Then State = 120

Rule no 33

If State is Z4 Then State = 150:

Rule no 34

If State is Z5 Then Zout - 10#

Rule no 35

If Error is LowSame and State is Z5 Then State = 140

Rule no 36

If Error is Higher and State is Z5 Then State = 0:

Rule no 37

If Zout is Higher Then Zout = 30#

Rule no 38

JMP 0

Rule diatas berlangsung secara iterasi , setiap satu siklus *state machine* dalam satu sumbu koordinat yang saling berkaitan satu dengan yang lainnya X,Y dan Z dan adanya kontrol umpan balik dari X ke Y ke Z kembali lagi ke X., dimana X merupakan suhu, Y merupakan kelembaban, dan Z merupakan tekanan.

Kenaikan suhu linier terhadap kenaikan kelembaban dan kenaikan tekanan sehingga kontrol utama terletak pada kontrol plant untuk suhu dibantu stabilisasi sistem oleh kontrol kelembaban dan kontrol tekanan.

Aksi kontrol stabilisasi sistem dilakukan secara iteratif berurutan dari X ke Y ke Z kemudian kontrol umpan balik *state machine* ke X loop tertutup dengan memperhatikan faktor dari supervisi berupa masukan *error* dan *D error*.

3.5. Perancangan perangkat lunak untuk Supervisi

Supervisi bertugas untuk mengolah data tranducer untuk dibandingkan terhadap setting point sehingga dapat diperoleh sinyal *error* dan *D error*, dirancang perangkat lunak untuk supervisi dengan menggunakan mikrokontroller 80C31BH.



BAB IV

PENGUJIAN DAN PENGUKURAN

BAB IV

PENGUJIAN DAN PENGUKURAN

Sebelum tugas akhir ini dijalankan sebagai sebuah sistem maka perlu dilaksanakan pengujian untuk tiap bagian sistemnya dan selanjutnya dilakukan kalibrasi dan pengukuran terhadap perangkat keras dan perangkat lunak yang ada.

4.1. Pengujian Alat

Pengujian dilakukan sebagai cara untuk mengetahui alat bekerja dengan baik atau tidak, menguji fungsi-fungsi yang telah direncanakan serta mengetahui kinerjanya.

Pengujian dilakukan terhadap modul yang telah dibuat yaitu :

1. Modul pengkondisi sinyal
2. Modul supervisi yang terdiri dari ADC dan DAC
3. Modul *fuzzy logic* Controller AL 220

Pengujian modul pengkondisi sinyal meliputi pengujian modul pengkondisi sinyal suhu, kelembaban dan tekanan.

Pengujian spervisi meliputi proses kerja dari supervisi untuk membaca ADC dan mengirim data ke DAC dan membandingkan dengan hasil perhitungan.

Pengujian modul *fuzzy logic* kontroller adalah dengan memberikan sinyal masukan pengkondisi sederhana atau disebut *fuzzy model* yang sederhana untuk melihat hasil respon sinyal keluarannya. Data bisa dilihat pada voltmeter digital.

4.2. Kalibrasi dan pengukuran

4.2.1. Suhu

Pengujian dan pengukuran dilakukan dengan membaca data pada keluaran sensor suhu tipe LM 35 produksi *National Semiconductor*. Hasil pengukuran dibandingkan terhadap termometer air raksa yang persisi.

Kalibrasi dilakukan dengan pembacaan pada multimeter digital dengan memasukkan pada air es untuk mendapatkan pembacaan mendekati nol dan pada air mendidih , untuk mendapatkan pembacaan pada suhu 100 °C kemudian keluaran di set pada 1 Volt.

Dari hasil pengukuran diatas terjadi kesalahan-kesalahan akibat ketidak linieran dari tranducer LM35 , kesalahan dalam pembacaan termometer gelas, dan kesalahan multimeter digital.

Tabel 4.1. Tabel Pengukuran Suhu

NO	Termometer Gelas (° C)	Keluaran Modul (mV)	Error (%)
1	40	0.39	2.5
2	50	0.49	2.0
3	60	0.60	0.0
4	70	0.69	1.42
5	80	0.79	1.25

4.2.2. Kelembaban

Pengujian dan pengukuran dilakukan dengan membaca data pada keluaran dua buah sensor suhu tipe LM 35 produksi National Semiconductor. Hasil pengukuran dibandingkan terhadap Hygrometer yang persisi buatan West German.

Kalibrasi dilakukan dengan pembacaan pada multimeter digital dengan memasukkan kondisi dengan kelembaban terukur dibandingkan terhadap Hygrometer dan data voltmeter digital.

Tabel 4.2. Tabel Pengukuran Kelembaban

NO	Hygrometer (%)	Keluaran Modul (%)	Error (%)
1	70	71	1.42
2	75	72	4
3	80	82	2.5
4	85	88	3.5
5	88	92	4.54

4.2.3. Tekanan

Pengujian dan pengukuran dilakukan dengan membaca data pada keluaran sensor infra merah yang dipasang pada *pipa u* untuk mengetahui kenaikan tekanan tiap Cm/Hg dengan menggunakan mikrokontroller 80C31BH. Kalibrasi dengan software dengan membandingkan dengan hukum fisika mengenai *pipa u* . Hukum mengenai *pipa u* adalah :

$$P = \rho \cdot g \cdot h$$

dimana:

P = Tekanan Terukur

ρ = massa jenis (air raksa)

h = tinggi permukaan Pipa U

dimana $P = 1 \text{ atm} = 76 \text{ cm /Hg}$

4.2.4. Modul Supervisi yang terdiri dari ADC dan DAC

Rangkaian supervisi diuji dengan membaca data masukan analog dari modul ADC MAX 158 dan dibandingkan antara nilai pengukuran dan perhitungan konversi analog ke digital, serta pemberian data pada modul DAC MAX 505 untuk dibandingkan adanya *error* konversi digital ke analog.

Terjadinya *error* disebabkan oleh linieritas ADC dan DAC itu sendiri, perancangan PCB yang kurang baik, ketelitian multimeter digital yang tidak sempurna.

Tabel 4.3. Tabel Pengujian Modul ADC

NO	Teori (mV)	Pengukuran (mV)	Error (%)
1	0	0	0
2	1000	1100	10
3	1500	1500	0
4	2000	1920	4
5	5000	4950	1

Tabel 4.4. Tabel Pengujian Modul DAC

NO	Data (BIN)	Perhitungan (mV)	Pengukuran (mV)	Err (%)
1	00000000	0	0	0
2	00001111	140	300	53.3
3	01111111	1240	1100	11.29
4	10000001	1250	1340	6.7
5	11111111	2490	2400	3.61

4.3. Pengujian modul *fuzzy logic* AL 220

Pengujian modul *fuzzy logic* AL220 dilakukan dengan cara sebagai berikut :

1. Melihat respon keluaran terhadap respon masukan sistem baik secara software dengan F.A.E 3.0 dan juga dengan memberikan uji data pada *fuzzy model*.
2. Membandingkan antara variabel fuzzy, aturan fuzzy, dengan file hasil kompilasi software F.A.E 3.0 yaitu file extension *.hex , disini akan terlihat bagaimana perangkat lunak fuzzy diterjemahkan menjadi isi dari memori internal pada memori AL 220.

4.3.1. Pengujian dan pengukuran pada respon AL 220

Pengujian modul *fuzzy logic* AL 220 dilakukan dengan cara melihat respon keluaran terhadap respon masukan sistem baik secara software dengan F.A.E 3.0 dan juga dengan memberikan uji data pada *fuzzy model*.

4.3.2. Perbandingan antara hasil rancangan dengan hasil kompilasi

Membandingkan antara variabel fuzzy, aturan fuzzy, dengan file hasil kompilasi software F.A.E 3.0 yaitu file extension *.hex , disini akan terlihat bagaimana perangkat lunak fuzzy diterjemahkan menjadi isi dari memori internal pada memori AL 220.

Tabel 4.4. Tabel Pembacaan Sistem hasil aksi *fuzzy logic*

No	Suhu (Celc)	Lembab (%)	Tekanan (Cm/Hg)	Waktu (Menit)
1	22	53	76	0
2	25	52	76	5
3	32	48	76	10
4	43	60	76	15
5	50	80	76	20

Tabel 4.5. Tabel pembacaan sistem dengan alat kalibrasi

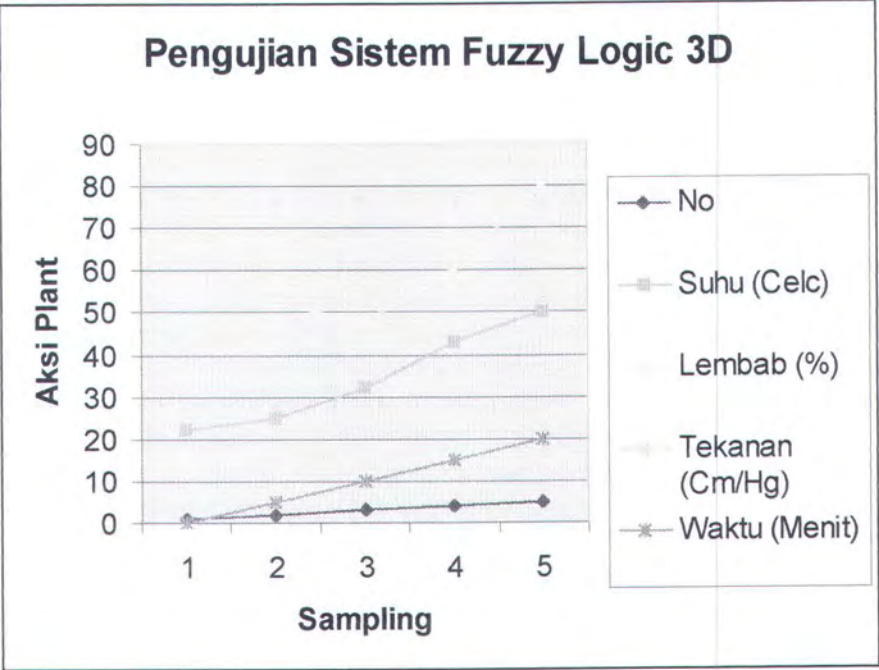
No	Suhu (Celc)	Lembab (%)	Tekanan (Cm/Hg)	Waktu (Menit)
1	25	60	76	0
2	25	58	76	5
3	32	60	76	10
4	44	70	76	15
5	52	78	76	20

Tabel 4.6. Tabel pembacaan keluaran tegangan *fuzzy logic*

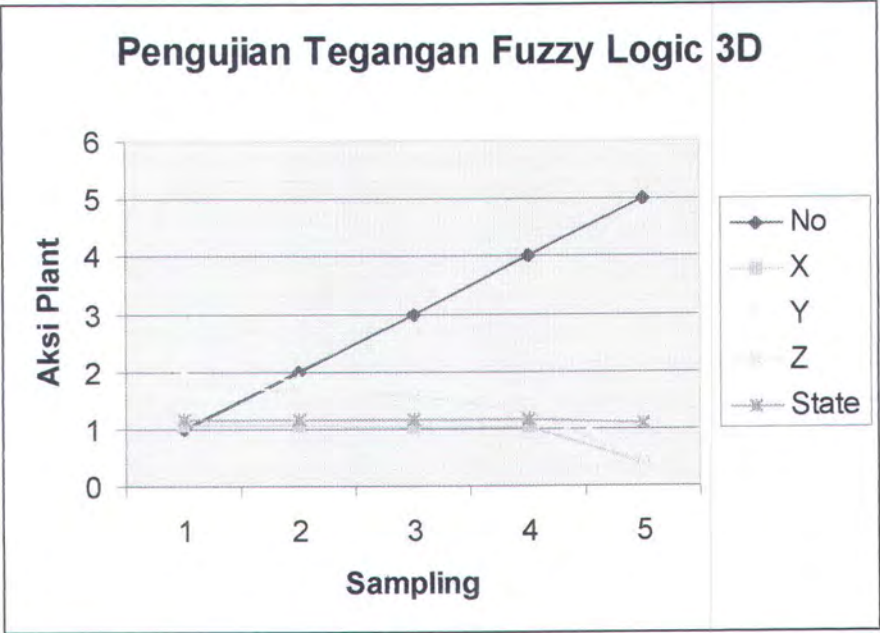
No	X (Volt)	Y (Volt)	Z (Volt)	State (Volt)
1	1.06	1.95	1.78	1.17
2	1.05	1.85	1.65	1.15
3	1.03	1.75	1.6	1.16
4	1.04	1.65	1.3	1.17
5	0.4	0.4	0.4	1.1

Tabel 4.7. Tabel error sistem

No	Error Suhu(%)	Error Lembab(%)	Error Tekanan(%)
1	8	11.6	0
2	0	10.3	0
3	0	20	0
4	2.27	14	0
5	3.85	2.5	0



Gambar 4.1. Grafik pengujian sistem



Gambar 4.2. Grafik pengujian tegangan fuzzy logic



BAB V

PENUTUP

BAB V

PENUTUP

5.1 Kesimpulan

Sesudah melakukan pembahasan mulai dari bab-bab sebelumnya dalam tugas akhir ini dapat dibuat kesimpulan :

1. Pembuatan kontrol otomatis dengan logika fuzzy perlu diketahui bagaimana perilaku sistem dan aturan-aturan fuzzy yang didasarkan pada pengalaman sistem.
2. Perancangan sensor untuk tekanan dengan hukum manometer gas udara dengan pipa u masih perlu di kalibrasi dengan sensor yang telah ada.
3. Proses iterasi tidak bisa terpenuhi secara bersamaan untuk mencapai kestabilan sistem.
4. Kestabilan sistem tidak mungkin tercapai secara bersamaan untuk tiap dimensi x, y, dan z.
5. Kestabilan sistem dikompensasi dengan adanya histeresis pada *rule* tiap dimensi.

5.2. Saran

Saran-saran yang diharapkan dapat berguna untuk pengembangan dan penggunaan lebih lanjut dari tugas akhir ini adalah :

1. Diperlukan pengembangan sistem dari *fuzzy logic* tiga dimensi menjadi N-Dimensi
2. Pengembangan dan penguasaan rule *fuzzy logic* tiga dimensi dapat menciptakan iterasi yang lebih baik.



DAFTAR PUSTAKA

DAFTAR PUSTAKA

1. Adaptive Logic, **AL220 Stand Alone Fuzzy logic Controller Data Sheet**, Basicconcepts , Inc
2. Coughlin , Robert F, dan Driscoll, Frederick F. , **Penguat Operasional dan Rangkaian Terpadu Linier** , Institut Teknologi Wentworth, Diterjemahkan oleh Herman Widodo Soemitro, Jakarta, Penerbit Airlangga ,1992
3. Jamshidi Mohammad dan Vadiiee, Nader dan Ross, Timothy J. , **Fuzzy logic and Control, Software dan Hardware Applications** , Prentice Hall 1993.
4. John Porges, M.I.Mech.E.,M.Inst,F, **Handbook of Heating Ventilating and Air Conditioning**, Seventh Edition , Newnes-Butterworths ,London ,Boston Sydney , Wllington, Durban, Toronto
5. J. Michael Jacob , **Industrial Control Electronic Application Design** , Prentice Hall International , Inc.
6. National Semiconductor , **1995 National Data Acquisition Databook** , USA , National Semiconductor , 1995
7. Steeman, J.P.M. , **Data Sheet Book 2** , Netherlands , Elektuur B.V. , Jakarta, P.T. Elex Media Komputindo , 1994
8. Team MGMP IPA KOTAMADIA Surabaya, Pelajaran Fisika (Kurikulum SMP 1994) kelas 1

9. William J.Mooney , **Optoelectronic Devices and Principles** , Monroe Community College , Prentice Hall International , Inc
10. ---, **Embeded Controller Hanbook data sheet** , Intel , Intel 1988
11. ---, **MAX154/MAX158 Data Sheet** , Sunnyvale, Maxim Integrated Products, 1999
12. ---, **MAX505/MAX506 Data Sheet** , Sunnyvale, Maxim Integrated Products, 1999
13. ---, **Motorola ,Optoelectronic Device Data, Global Opto isolator Product**, Motorola , Inc



LAMPIRAN

LAMPIRAN A : PERANGKAT LUNAK *FUZZY LOGIC* FAE 3.0

FILE ST02.INP

50	60	100	50	40	100
50	60	100	50	40	100
40	50	100	50	40	100
40	50	100	50	40	100
40	50	100	50	40	100
40	50	100	50	40	100
40	50	100	50	40	100
40	50	100	50	40	100
40	50	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
30	40	100	50	40	100
50	40	100	50	40	100
50	40	100	50	40	100

FILE ST02.FLS

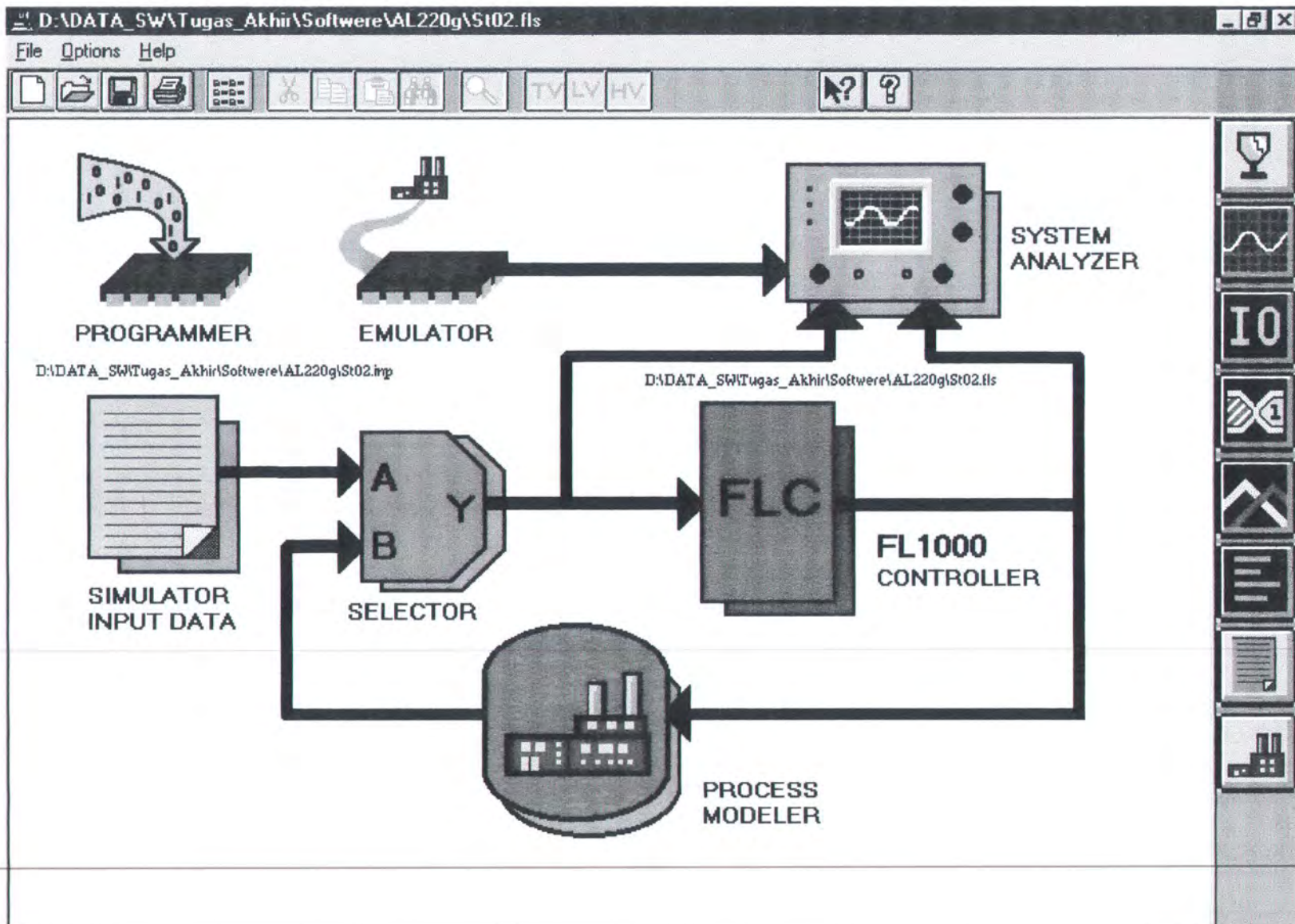
```
;
;
VARIABLES
;
;
;
INPUTS
;
Error = 0
ErrDly = 0
Setpoint = 0
Ain4 = 0
;
OUTPUTS
;
State = 0
Xout = 0
Yout = 0
Zout = 0
;
TERMS
;
Error is Higher ( ErrDly , 0 , 1 , LE )
Error is LowSame ( ErrDly , 0 , 1 , LI )
Error is NotZero ( 0 , 0 , 1 , RI )
Error is Zero ( 0 , 0 , 1 , SI )
Setpoint is Off ( 0 , 0 , 1 , SI )
Setpoint is On ( 0 , 0 , 1 , RI )
State is X1 ( 10 , 0 , 1 , SI )
State is X2 ( 20 , 0 , 1 , SI )
State is X3 ( 30 , 0 , 1 , SI )
State is X4 ( 40 , 0 , 1 , SI )
State is X5 ( 50 , 0 , 1 , SI )
State is Y1 ( 60 , 0 , 1 , SI )
State is Y2 ( 70 , 0 , 1 , SI )
State is Y3 ( 80 , 0 , 1 , SI )
State is Y4 ( 90 , 0 , 1 , SI )
State is Y5 ( 100 , 0 , 1 , SI )
State is Z1 ( 110 , 0 , 1 , SI )
State is Z2 ( 120 , 0 , 1 , SI )
State is Z3 ( 130 , 0 , 1 , SI )
State is Z4 ( 140 , 0 , 1 , SI )
State is Z5 ( 150 , 0 , 1 , SI )
State is Zero ( 0 , 0 , 1 , SI )
Xout is Higher ( 55 , 5 , 1 , RI )
Yout is Higher ( 127 , 5 , 1 , RI )
Zout is Higher ( 127 , 5 , 1 , RI )
```



```

;
RULES
;
If Setpoint is Off Then Xout = 0#
If Setpoint is On and State is Zero Then State = 10#
If Error is Zero and State is X1 Then State = 0
If State is X1 Then State = 20:
If State is X2 Then Xout + 1#
If State is X2 Then State = 30#
If Error is LowSame and State is X3 Then State = 40
If Error is Higher and State is X3 Then State = 20
If State is X4 Then State = 50:
If State is X5 Then Xout - 1#
If Error is LowSame and State is X5 Then State = 40
If Error is Higher and State is X5 Then State = 60:
If Xout is Higher Then Xout = 30#
If Error is Zero and State is Y1 Then State = 0
If State is Y1 Then State = 70:
If State is Y2 Then Yout + 40#
If State is Y2 Then State = 80#
If Error is LowSame and State is Y3 Then State = 90
If Error is Higher and State is Y3 Then State = 70
If State is Y4 Then State = 100:
If State is Y5 Then Yout - 20#
If Error is LowSame and State is Y5 Then State = 90
If Error is Higher and State is Y5 Then State = 110:
If Yout is Higher Then Yout = 30#
If Error is Zero and State is Z1 Then State = 0
If State is Z1 Then State = 120:
If State is Z2 Then Zout + 20#
If State is Z2 Then State = 130#
If Error is LowSame and State is Z3 Then State = 140
If Error is Higher and State is Z3 Then State = 120
If State is Z4 Then State = 150:
If State is Z5 Then Zout - 10#
If Error is LowSame and State is Z5 Then State = 140
If Error is Higher and State is Z5 Then State = 0:
If Zout is Higher Then Zout = 30#
JMP 0
;
TABLES
;
;
END

```



TERM Editor

TERMS List

State is X5 (50 , 0 , 1 , SI)
State is Y1 (60 , 0 , 1 , SI)
State is Y2 (70 , 0 , 1 , SI)
State is Y3 (80 , 0 , 1 , SI)
State is Y4 (90 , 0 , 1 , SI)
State is Y5 (100 , 0 , 1 , SI)

TERM Definition

Data Name: State
MF Name: Y3
Center, Any, Bit: 80
Width: 0
Slope: 1
Type: Symetric Inclusive

Add Change Remove

Apply Cancel OK

TABLE Editor

INPUT NAMES

Error
ErrDly
Setpoint
Ain4

OUTPUT NAMES

State
Xout
Yout
Zout

VARIABLE NAMES

Table Edit Apply Cancel OK

Simulation Control

0 D:\DATA_SW\Tugas_Akhir\Software\AL220g\St02.flx 1999

RUN BATCH STEP BREAK STOP

Error=0
ErrDly=0
Setpoint=0
Ain4=0
State=0
Xout=0
Yout=0
Zout=0

Force To:

Force

☒ This Sample
☐ All Samples

Batch File Name

Add File Remove File

ADRS 0 MIN 0 MAX 0

```

* 0 If Setpoint is Off Then Xout = 0#
4 If Setpoint is On and State is Zero Then State = 10#
10 If Error is Zero and State is X1 Then State = 0
15 If State is X1 Then State = 20:
19 If State is X2 Then Xout + 1#
23 If State is X2 Then State = 30#
27 If Error is LowSame and State is X3 Then State = 40
32 If Error is Higher and State is X3 Then State = 20
37 If State is X4 Then State = 50:
41 If State is X5 Then Xout - 1#
45 If Error is LowSame and State is X5 Then State = 40
50 If Error is Higher and State is X5 Then State = 60:
56 If Xout is Higher Then Xout = 30#
60 If Error is Zero and State is Y1 Then State = 0
65 If State is Y1 Then State = 70:
69 If State is Y2 Then Yout + 40#
73 If State is Y2 Then State = 80#
77 If Error is LowSame and State is Y3 Then State = 90
82 If Error is Higher and State is Y3 Then State = 70
87 If State is Y4 Then State = 100:
91 If State is Y5 Then Yout - 20#
95 If Error is LowSame and State is Y5 Then State = 90
100 If Error is Higher and State is Y5 Then State = 110:
106 If Yout is Higher Then Yout = 30#
  
```

D:\DATA_SW\Tugas_Akhir\Software\AL220g\St02.flx - [Histogram Viewer]

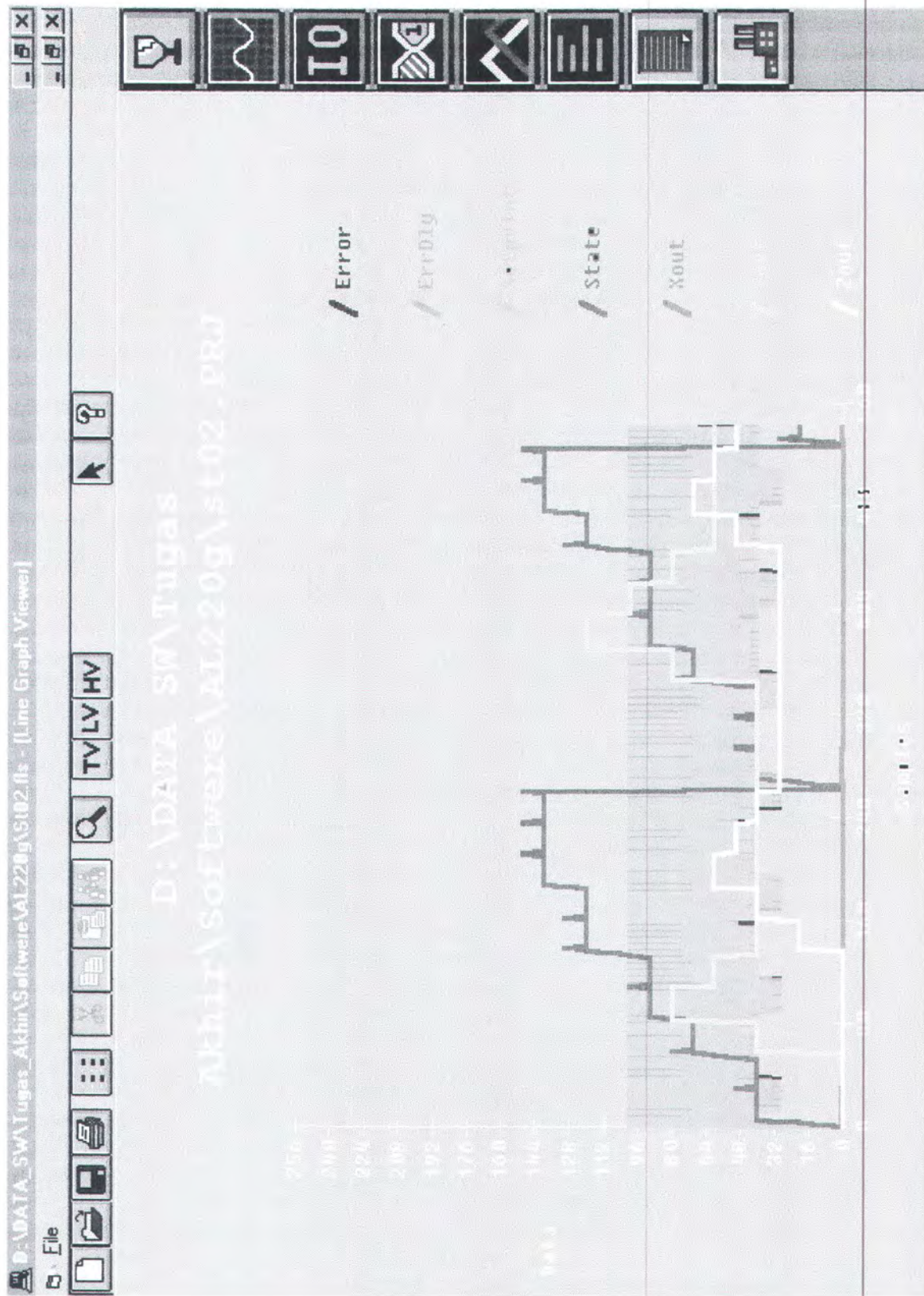
File

TV LV HV

Address	Count	Rule
0	1	If Setpoint is Off Then Xout = 0#
4	8	If Setpoint is On and State is Zero Then State = 10#
10	0	If Error is Zero and State is X1 Then State = 0
15	8	If State is X1 Then State = 20:
19	10	If State is X2 Then Xout + 1#
23	10	If State is X2 Then State = 30#
27	8	If Error is LowSame and State is X3 Then State = 40
32	2	If Error is Higher and State is X3 Then State = 20
37	18	If State is X4 Then State = 50:
41	18	If State is X5 Then Xout - 1#
45	11	If Error is LowSame and State is X5 Then State = 40
50	7	If Error is Higher and State is X5 Then State = 60:
56	0	If Xout is Higher Then Xout = 30#
60	0	If Error is Zero and State is Y1 Then State = 0
65	7	If State is Y1 Then State = 70:
69	16	If State is Y2 Then Yout + 40#
73	16	If State is Y2 Then State = 80#
77	7	If Error is LowSame and State is Y3 Then State = 90
82	9	If Error is Higher and State is Y3 Then State = 70
87	18	If State is Y4 Then State = 100:
91	18	If State is Y5 Then Yout - 20#
95	11	If Error is LowSame and State is Y5 Then State = 90
100	7	If Error is Higher and State is Y5 Then State = 110:
106	3	If Yout is Higher Then Yout = 30#
110	0	If Error is Zero and State is Z1 Then State = 0
115	7	If State is Z1 Then State = 120:

Simulate the Control System.

10:31 PM 1/24/00



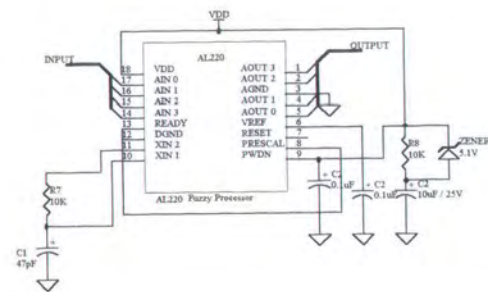


History Table View

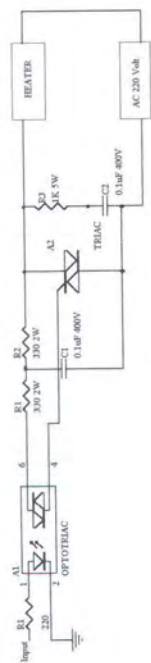
File Options

	19 152	77	100	30	40	40	40	2	0	10
	19 18	91	100	30	40	40	40	2	0	10
	19 188	95	30	40	100	40	40	2	0	10
	19 21	106	30	40	100	40	40	2	0	10
	19 218	110	40	100	30	40	40	2	0	10
	19 236	119	40	100	30	40	40	2	0	10
	19 244	123	100	30	40	40	40	2	0	10
	19 252	127	30	40	100	40	40	2	0	10
	19 29	141	30	40	100	40	40	2	0	10
	19 288	145	40	100	30	40	40	2	0	10
	19 31	158	40	100	30	40	40	2	0	10
	19 318	160	100	30	40	40	40	2	0	10
	19 32	0	100	30	40	40	40	2	0	10
	19 328	4	30	40	100	40	40	2	0	10
	19 34	10	40	100	30	40	40	2	0	10
	19 358	19	40	100	30	40	40	2	0	10
	19 366	23	100	30	40	40	40	2	0	10
	19 374	27	30	40	100	50	50	2	0	10
	19 402	41	30	40	100	50	50	1	0	10
	19 41	45	40	100	30	40	40	1	0	10
	19 432	55	40	100	30	40	40	1	0	10
	19 44	60	100	50	40	40	40	1	0	10
	19 458	63	100	50	40	40	40	1	0	10
	19 465	73	50	40	100	40	40	1	0	10

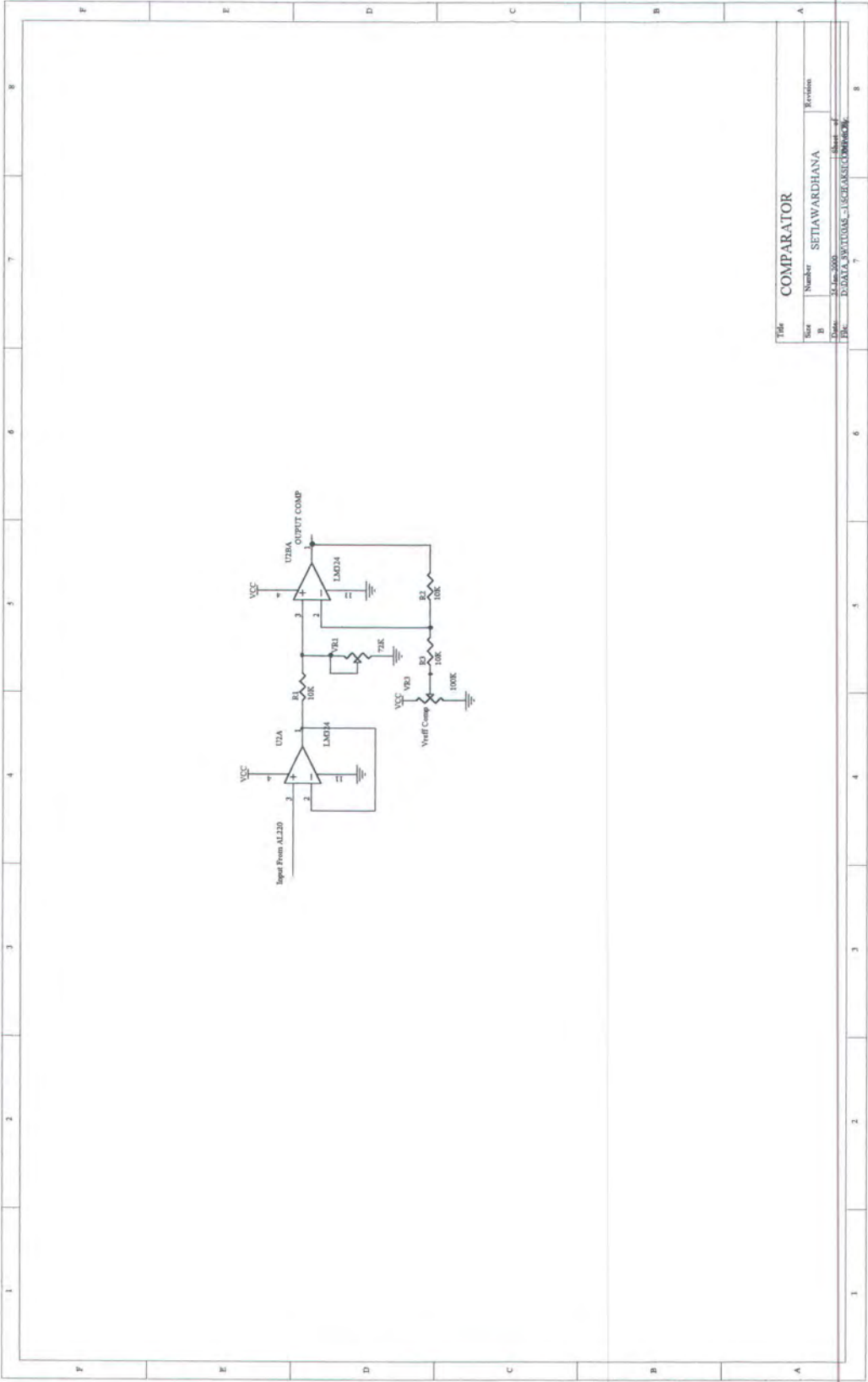
LAMPIRAN B : SKEMA RANGKAIAN



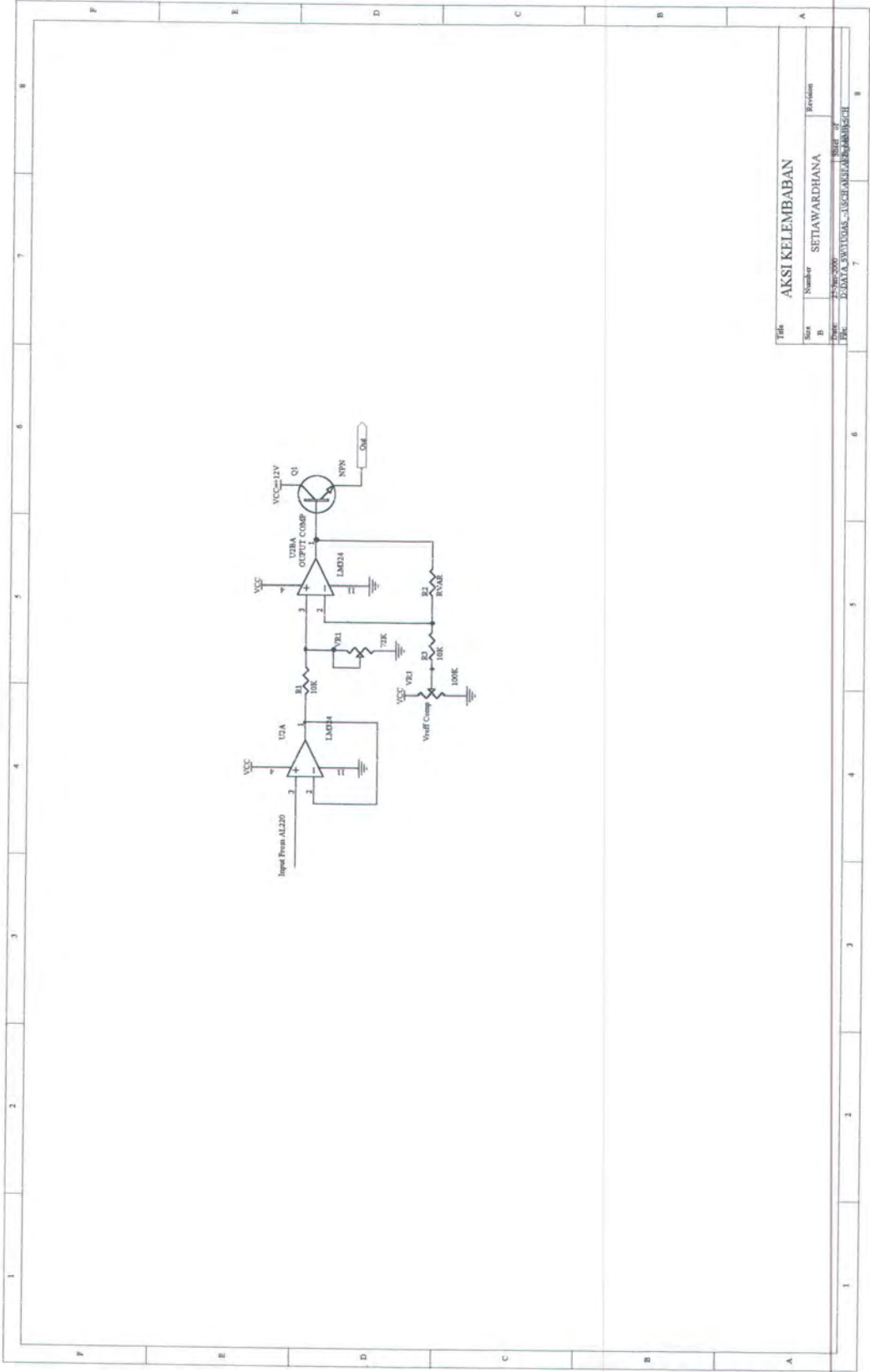
Title			
AL220 FUZZY LOGIC BOARD			
Size	Number	Revision	
B	SETIAWARDHANA		
Date	6-Feb-2000	Sheet of	
Title: D:\DATA\KIRITUGAS_1\SCHEMATA\AL220\FUZZY			



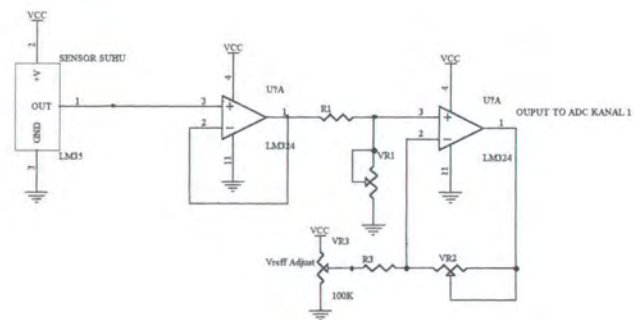
Title		Revision	
Size	Number	SETIAWARDHANA	
Date	6 Feb 2000	Sheet of	
D:\DATA_2000\Tugas\1450444664\1450444664.dwg			



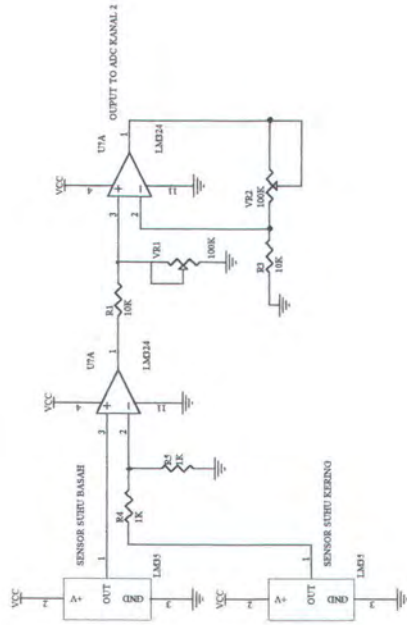
Title		COMPARATOR	
Size	Number	SETIAWARDHANA	
B			
Date	14 Jan 2020	Sheet 1 of 1	
File	D:\ATA\KUTUKAN\3\ESKAS\COMPARATOR		



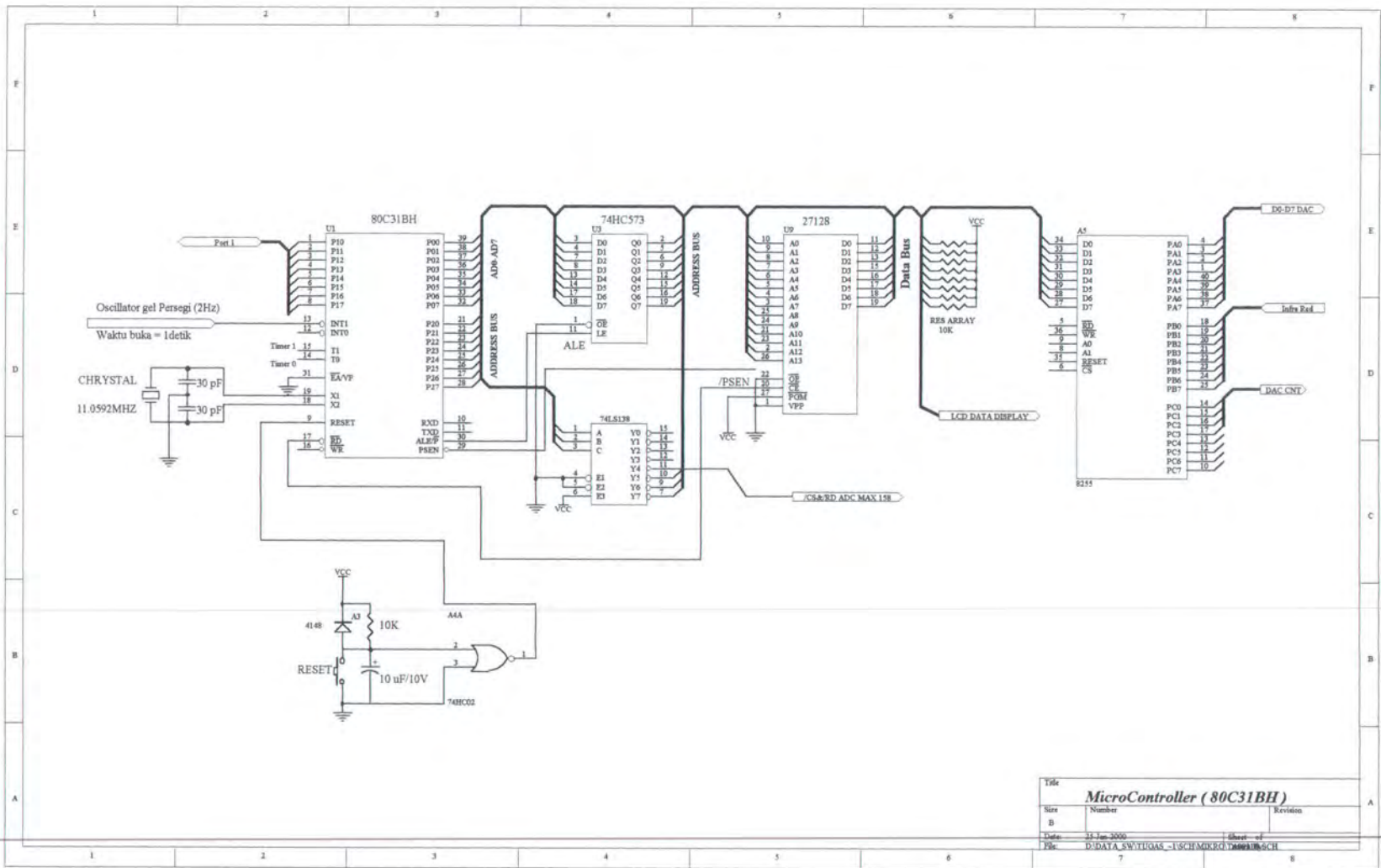
Title		AKSI KELEMBABAN	
Size	Number	Setiawardhana	Revision
D	1		
Date		15-05-2024	
Page		1	

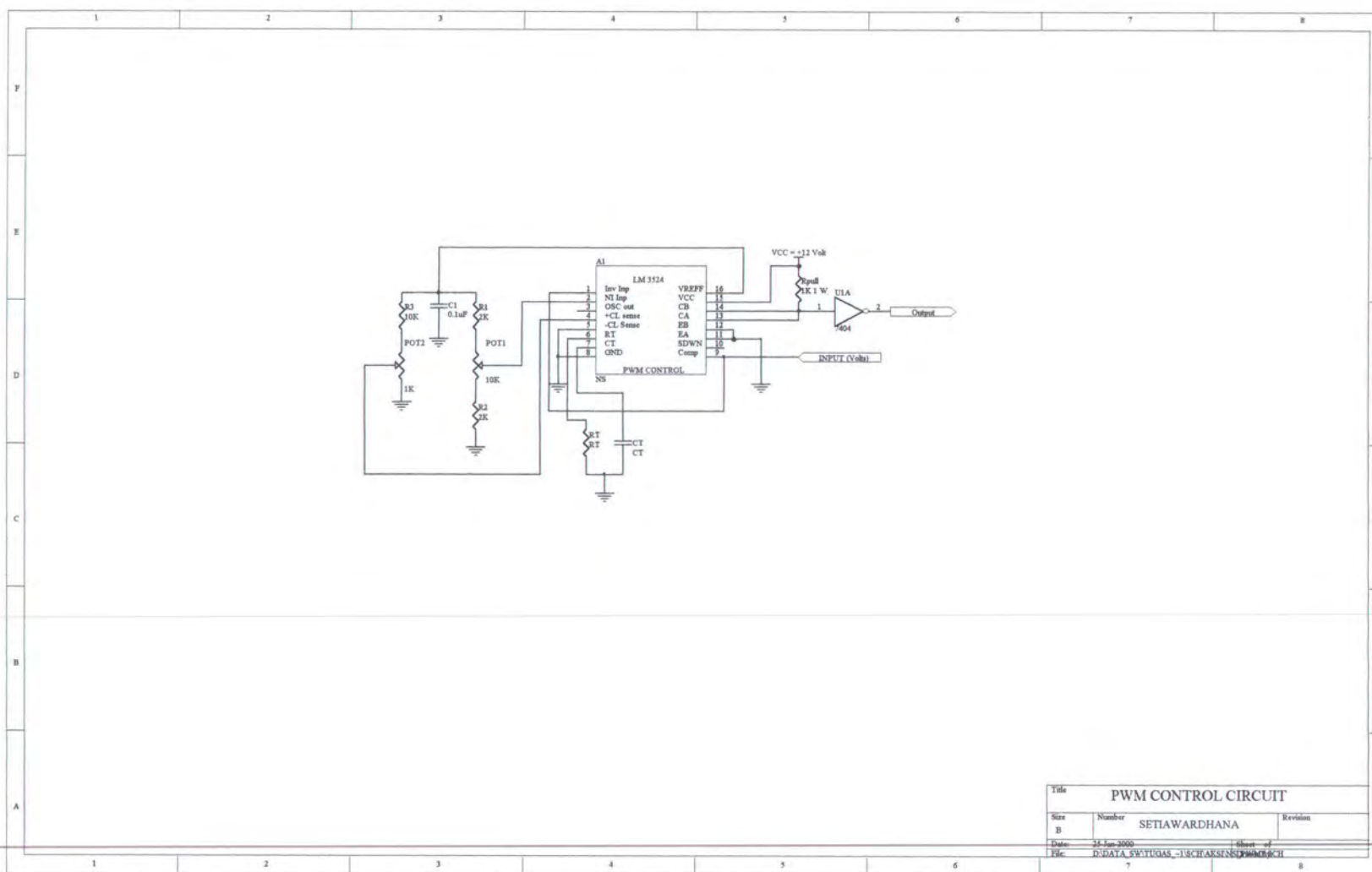


Title		
DC PROCESSING TEMPERATURE		
Size	Number	Revision
B	SETIAWARDHANA	
Date:	25-Jan-2020	Sheet of 1
File:	D:\DATA_SW\TUGAS_1\SCH\SENSOR	Sheet 11



Ttd: DC PROCESSING KELEMBABAN			
No		Setiawardhana	
B		Kendari	
Date: 24 Jan 2025		1 sheet of 1	
File: E:\DATA\WOTUMAS_2\UGRSENSEOR\HARDWARE\GCU			





Title			PWM CONTROL CIRCUIT
Size	Number	Revision	
B	SETIAWARDHANA		
Date:	24 Jan 2000	Sheet of	
File:	D:\DATA_SW\TUGAS - I\SCH ARSIN\KIRK\KIRK		

LAMPIRAN C : PERANGKAT LUNAK MIKROKONTROLLER

ORG 0000H
AJMP MULAI

ORG 0050H

; ---- KONSTANTA UNTUK LCD ----

DISPCLR EQU 00000001B
FUNCSET EQU 00111000B;INTERFACE DATA LENGTH : 8 BITS
ENTRMOD EQU 00000110B;INCREMENT, NO DISPLAY SHIFT
DISPON EQU 00001100B;DISPLAY ON, CURS OFF, BLNK OFF
CURSOR EQU 00001110B;DISPLAY ON, CURS ON, BLINK OFF
BLINK EQU 00001101B;DISPLAY ON, CURSOR OFF, BLNK ON
GESER EQU 00011000B ;GESER LCD
HOME EQU 00000010B ;POSISI KURSOR DIAWAL LCD

; ---- ADDRESS UNTUK LCD ----

LCD EQU 0A000H
LCD0 EQU LCD+0 ;LCD CONTROL OPERATION
LCD1 EQU LCD+1 ;LCD DATA OPERATION
ADC EQU 8000H ;ADC ADDRESS
ADC0 EQU ADC+0 ;ADC KANAL 1
ADC1 EQU ADC+1 ;ADC KANAL 2
ADC2 EQU ADC+2 ;ADC KANAL 3
BLANK EQU 4000H
PPI EQU 0E000H ;PPI8255
PA EQU PPI+0
PB EQU PPI+1
PC EQU PPI+2
PCTRL EQU PPI+3

; ---- KALIMAT YANG DITAMPILKAN ----

KAL1: DB ' FUZZY LOGIC 3D ',0
KAL2: DB 'TUGAS AKHIR 1999/2000 ',0
KAL3: DB ' Dosen Pembimbing Ir SOETIKNO ',0
KAL4: DB 'By-SETIAWARDHANA (2295.100.029) ',0
KAL5: DB ' ',0
KAL6: DB ' ',0
KAL7: DB 'Suhu:',0
KAL8: DB 'Humidity:',0
KAL9: DB 'Press:',0
KAL10: DB 'Set:',0
KAL11: DB 'Err:',0
KAL12: DB 'C',0
KAL14: DB 'Cm/Hg',0
KAL15: DB 'ERROR = ',0
KAL16: DB 'D ERR = ',0
KAL17: DB 'Set Suhu = ',0
KAL18: DB 'Set Lembab = ',0
KAL19: DB 'Set Tekanan= ',0

```

KAL20: DB ' MAXIMUM LEVEL',0
KAL21: DB ' MINIMUM LEVEL',0
;--- VARIABEL DATA PROSES ----
SET_FUZ EQU 66H
SETX EQU 67H
SETY EQU 68H
SETZ EQU 69H
INFRA EQU 70H
ERR_1 EQU 71H
ERR_2 EQU 72H
ERR_3 EQU 73H
ERROR EQU 74H
D_ERR EQU 75H
TOMBOL EQU 76H
SUHU EQU 77H
TEKANAN EQU 78H
LEMBAB EQU 79H

```

```

ORG 500H

```

```

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;      MAIN PROGRAM
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MULAI:

```

```

CLR A
LCALL DLY_PPI
LCALL INIT_PPI
LCALL DLY_PPI
LCALL INIT_LCD

```

```

MOV ERR_1,#00H
MOV ERR_2,#00H
MOV ERR_3,#00H

```

```

MOV ERROR,#00H
MOV D_ERR,#00H
MOV SET_FUZ,#00H

```

```

MOV A,#76
MOV TEKANAN,A

```

```

MOV P1,#00H      ;RESET ISI PORT A 0000 0000
MOV TOMBOL,#0    ;RESET TOMBOL KE 0

```

```

MOV SETX,#50D
MOV SETY,#80D
MOV SETZ,#77D

```

CLR A

; ----- POOLLING SETELAH ADA KEYPAD DITEKAN -----

POOLING:

;----- baca suhu dan lembab -----
LCALL ADC158

;----- baca tekanan pipa u -----
LCALL BACA_PIPA_U

;----- error -----
LCALL ERROR_SUHU
LCALL ERROR_LEMBAB
LCALL ERROR_TEGANAN
LCALL HITUNG_ERROR

;----- beri data ke DAC -----
LCALL TO_DAC

;----- update last err=err-----
LCALL UPDT

;----- cek tombol ditekan -----
MOV A,P1
ANL A,#01H
CJNE A,#01H,TIDAK_DITEKAN
MOV A,TOMBOL
ADD A,#1
MOV TOMBOL,A
CJNE A,#9,TUNGGU_DILEPAS

MOV TOMBOL,#0 ;RESET TOMBOL KE 0
LCALL DISP_CLR

;----- bouncing effect -----

TUNGGU_DILEPAS:

MOV A,P1
ANL A,#01H
LCALL DLAY
CJNE A,#0,TUNGGU_DILEPAS

TIDAK_DITEKAN:

MOV A,TOMBOL
CJNE A,#0,BUKAN_TAMPILAN_AWAL

LCALL NDISP ;DISPLAY AWAL

```
MOV A,#GESER      ;PUTAR DISPLY
LCALL CONTROLOUT
LCALL DELAY2
AJMP POOLING
```

```
BUKAN_TAMPILAN_AWAL:
  CJNE A,#1,BUKAN_SUHU
  LCALL DISP_CLR
  LCALL SUHUKU      ;SUHU
  AJMP POOLING
```

```
BUKAN_SUHU:
  CJNE A,#2,BUKAN_LEMBAB
  LCALL DISP_CLR
  LCALL LEMBABKU    ;LEMBAB
  AJMP POOLING
```

```
BUKAN_LEMBAB:
  CJNE A,#3,BUKAN_TEGANAN
  LCALL DISP_CLR
  LCALL TEGANANKU   ;TEGANAN
  AJMP POOLING
```

```
BUKAN_TEGANAN:
  CJNE A,#4,SETTING_X
  LCALL DISP_CLR
  LCALL ERR_DISP    ;ERROR SISTEM
  AJMP POOLING
```

```
SETTING_X:
  CJNE A,#5,SETTING_Y
  LCALL DISP_CLR
  LCALL SET_SUHU
  AJMP POOLING
```

```
SETTING_Y:
  CJNE A,#6,SETTING_Z
  LCALL DISP_CLR
  LCALL SET_LEMBAB
  AJMP POOLING
```

```
SETTING_Z:
  CJNE A,#7,CREATED
  LCALL DISP_CLR
  LCALL SET_TEGANAN
  AJMP POOLING
```

```
CREATED:
```



```

CJNE A,#8,KE_AWAL
LCALL TA_CREATED
MOV A,#GESER      ;PUTAR DISPLY
LCALL CONTROLOUT
LCALL DELAY2
AJMP POOLING

```

```

KE_AWAL:AJMP POOLING

```

```

;----- LCD -----
POSISI2.1:
    MOV A,R1      ;KOLOM KE
POSISI2:
    ADD A,#11000000B ;POSISI DI BARIS 2
    SJMP POSISI.SUB
;
;
POSISI1.1:
    MOV A,R0      ;KOLOM KE
POSISI1:
    ADD A,#10000000B ;POSISI DI BARIS 1
POSISI.SUB:
    DEC A          ;AWALAN POSISI KOLOM DIMULAI DARI KOLOM KE
    LCALL CONTROLOUT ;KIRIM SEBAGAI OPERASI CONTROL
    RET
;
;
PRINTSTRING2:                ;CETAK STRING DI BARIS 2 KOLOM 1
    LCALL POSISI2.1          ;BARIS 2 KOLOM1
    SJMP PRINTSTRING         ;CETAK STRING
;
;
PRINTSTRING1:                ;CETAK STRING DI BARIS 1 KOLOM 1
    LCALL POSISI1.1          ;BARIS 1 KOLOM 1
;
;
PRINTSTRING:                 ;CETAK STRING
    SJMP OUTSTRING           ;AMBIL DATA DULU
PRINTSTRINGLOOP:
    LCALL DATAOUT            ;KIRIM SEBAGAI OPERASI DATA
    INC DPTR                 ;POSISI DATA BERIKUTNYA
;
;
OUTSTRING:
    CLR A                    ;POINTER = 0
    MOVC A,@A+DPTR           ;AMBIL DTA DI EPROM BERDASARKAN
DPTR
    JNZ PRINTSTRINGLOOP      ;MASIH ADA DATA ?
    RET
;
;
CONTROLOUT:
    PUSH DPH                 ;SIMPAN DPH DI STACK

```

```

PUSH DPL          ;SIMPAN DPL DI STACK
MOV DPTR,#LCD0    ;ALAMAT OPERASI CONTROL LCD
SJMP LCD.OUT      ;KIRIM KE LCD

```

```

;
DATAOUT:

```

```

    PUSH DPH      ;SIMPAN DPH DI STACK
    PUSH DPL      ;SIMPAN DPL DI STACK
    MOV DPTR,#LCD1 ;ALAMAT OPERASI DATA LCD

```

```

LCD.OUT:

```

```

    MOVX @DPTR,A    ;KIRIM KE LCD

```

```

;
DELAY.LCD:

```

```

    MOV A,#250
    DJNZ ACC,$
    POP DPL      ;AMBIL KEMBALI DPH DARI STACK
    POP DPH      ;AMBIL KEMBALI DPL DARI STACK
    RET

```

```

;
DELAY.INIT.LCD:

```

```

    MOV R6,#20H

```

```

DLY.LCD.LP:

```

```

    MOV R7,#0
    DJNZ R7,$
    DJNZ R6,DLY.LCD.LP
    RET

```

```

;-----

```

```

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

;    PROCEDURE INIT LCD

```

```

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

INIT_LCD:

```

```

    MOV A,#DISPCLR    ;DISPLAY CLEAR

```

```

    LCALL CONTROLOUT

```

```

    LCALL DELAY.INIT.LCD

```

```

    MOV A,#FUNCSET    ;FUNCTION SET

```

```

    LCALL CONTROLOUT

```

```

    LCALL DELAY.INIT.LCD

```

```

    MOV A,#DISPON     ;DISPLAY ON

```

```

    LCALL CONTROLOUT

```

```

    LCALL DELAY.INIT.LCD

```

```

    MOV A,#ENTRMODE   ;ENTRY MODE

```

```

    LCALL CONTROLOUT

```

```

    LCALL DELAY.INIT.LCD

```

```

    RET

```

```

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;      PROCEDURE DISPCLR LCD
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

DISP_CLR:
    MOV A,#DISPCLR      ;DISPLAY CLEAR
    LCALL CONTROLOUT
    LCALL DELAY.INIT.LCD
    RET

```

```

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;      DELAY GESER LCD
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

DELAY:
    MOV R6,#01FH
DLY1:  MOV R7,#0FFH
        DJNZ R7,$
        DJNZ R6,DLY1
        RET

```

```

DELAY2:
    LCALL DELAY
    LCALL DELAY
    LCALL DELAY
    RET

```

```

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;      PRODCEDURE DISPLAY AWAL
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

NDISP:
    MOV R0,#1D
    MOV DPTR,#KAL1
    LCALL PRINTSTRING1
    MOV R1,#1D
    MOV DPTR,#KAL2
    LCALL PRINTSTRING2
    RET

```

```

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;      CREATED TUGAS AKHIR 1999/2000
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

TA_CREATED:
    MOV R0,#1D      ;TULIS KAL3
    MOV DPTR,#KAL3
    LCALL PRINTSTRING1
    MOV R1,#1D      ;TULIS KAL4
    MOV DPTR,#KAL4
    LCALL PRINTSTRING2
    RET

```



```
;%%%%%%%%%
; READ ADC AND DELAY MAX158
```

```
;%%%%%%%%%
```

```
; cara baca :
```

- ; 1. /cs dan /rd low ,delay beberapa saat
- ; 2. Baca data dari D0-D7
- ; 3. Set /cs dan /rd high dan ulangi no 1
- ; 4. Posisi pin hardware pada AD0-AD7 dan D0-D7

```
DLAY:
```

```
MOV R6,#01FH
```

```
DLY: MOV R7,#0FFH
```

```
DJNZ R7,$
```

```
DJNZ R6,DLY
```

```
RET
```

```
ADC158:
```

```
;----- baca adc kanal 1 -----
```

```
MOV DPTR,#ADC0 ;KANAL 1
```

```
LCALL DLY
```

```
MOVX A,@DPTR
```

```
MOV LEMBAB,A
```

```
MOV DPTR,#BLANK ;/RD=/CS=HIGH
```

```
;----- baca adc kanal 2 -----
```

```
MOV DPTR,#ADC1 ;KANAL 2
```

```
LCALL DLY
```

```
MOVX A,@DPTR
```

```
MOV SUHU,A
```

```
MOV DPTR,#BLANK ;/RD=/CS=HIGH
```

```
RET
```

```
;%%%%%%%%%
```

```
; PROCEDURE GIVE ERROR AND DLY ERROR DAC MAX505
```

```
;%%%%%%%%%
```

```
; cara kirim data :
```

- ; 1. /LDAC = 0 dan /WR = 0 (PC3= /WR) /LDAC=GND
- ; 2. Error = Kanal 1 DError = Kanal 2
- ; 3. Pin /CS untuk ADC MAX158 = 1 (Tidak Aktif)
- ; 4. Pada hardware posisi pin AD0-AD7 dan D0-D7

```
TO_DAC:
```

```
;----- PPI8255 -----
```

```
;----- PC3 PC2 PC1 PC0 -----
```

```
;----- /LDAC /WR A1 A0 -----
```

```
LCALL DLY_PPI
```



```
CLR A
MOV A,ERROR      ;err
MOV DPTR,#PA
MOVX @DPTR,A
LCALL DLY_PPI
```

```
MOV A,#00000000B ;wr=0 kanal 1 ldac = 0
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI
```

```
MOV A,#00000100B ;wr=1 ldac = 0
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI
```

```
CLR A
MOV A,D_ERR      ;err dly
MOV DPTR,#PA
MOVX @DPTR,A
LCALL DLY_PPI
```

```
MOV A,#00000001B ;wr=0 kanal 2 ldac =0
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI
```

```
MOV A,#00000101B ;wr=1 ldac = 0
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI
```

```
MOV A,SET_FUZ    ;setting point
MOV P1,A
```

```
RET
```

```
;-----
DLY_PPI:
    MOV B,#2
LOOP:
    MOV A,#0FFH
    DJNZ A,$
    DJNZ B,LOOP
    RET
```

```
;-----
INIT_PPI:
```

;--- PA=OUT PB=IN PCLOW=OUT PCHIGH=OUT ---

```
MOV A,#10000010B    ;82H
MOV DPTR,#PCTRL
MOVX @DPTR,A
RET
```

```
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;    PROCEDURE OLAH DATA KE DESIMAL
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
OLAHD1:
```

;--cari bentuk desimalnya--

```
MOV B,#100D
DIV AB
MOV R3,A    ;dapat RATUSAN
MOV A,B
MOV B,#10D
DIV AB
MOV R4,A    ;dapat PULUHAN
MOV R5,B    ;dapat SATUAN
```

```
MOV A,R3    ;tampil RATUSAN
LCALL WR_DT
```

```
MOV A,R4    ;tampil PULUHAN
LCALL WR_DT
```

```
MOV A,R5    ;tampil SATUAN
LCALL WR_DT
```

RET

```
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;    PROCEDURE WRITE BILANGAN KE LCD
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DLAY02:
```

```
    MOV R6,#01H
DLY02: MOV R7,#250
    DJNZ R7,$
    DJNZ R6,DLY02
    RET
```

WR_DT:

```
    PUSH DPH
    PUSH DPL
    MOV DPTR,#LCD1
    ADD A,#30H
```

```
MOVX @DPTR,A
LCALL DLAY02
POP DPL
POP DPH
RET
```

```
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;  PROCEDURE TAMPILKAN SUHU DI LCD
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SUHUKU:
```

```
MOV R0,#1D      ;TULIS KAL7
MOV DPTR,#KAL7
LCALL PRINTSTRING1
```

```
MOV A,SUHU
LCALL OLAHDT
```

```
MOV A,#0DFH      ;tampil DERAJAT
SUBB A,#30H
LCALL WR_DT
```

```
MOV R0,#10D      ;TAMPILKAN HURUF C (Celcius)
MOV DPTR,#KAL12
LCALL PRINTSTRING1
```

```
MOV R1,#1D      ;TULIS SET SUHU
MOV DPTR,#KAL10
LCALL PRINTSTRING2
```

```
MOV A,SETX      ;SET
LCALL OLAHDT
```

```
MOV R1,#10D      ;TULIS ERROR SUHU-SETTING
MOV DPTR,#KAL11
LCALL PRINTSTRING2
```

```
MOV A,ERR_1      ;ERR
LCALL OLAHDT
```

```
RET
```

```
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;  PROCEDURE TAMPILKAN HUMIDITY DI LCD
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
LEMBABKU:
```

```
MOV R0,#1D      ;TULIS KAL8
```

```
MOV DPTR,#KAL8
LCALL PRINTSTRING1
```

```
MOV A,LEMBAB
LCALL OLAHDT
```

```
MOV A,#025H      ;tampil percent
SUBB A,#30H
LCALL WR_DT
```

```
MOV R1,#1D      ;TULIS SET KELEMBABAN
MOV DPTR,#KAL10
LCALL PRINTSTRING2
```

```
MOV A,SETY      ;SET
LCALL OLAHDT
```

```
MOV R1,#10D      ;TULIS ERR LEMBAB-SETTING
MOV DPTR,#KAL11
LCALL PRINTSTRING2
```

```
MOV A,ERR_2      ;ERR
LCALL OLAHDT
```

```
RET
```

```
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;      PROCEDURE TAMPILKAN PRESSURE DI LCD
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TEKANANKU:
```

```
MOV R0,#1D      ;TULIS KAL9
MOV DPTR,#KAL9
LCALL PRINTSTRING1
```

```
MOV A,TEKANAN
LCALL OLAHDT
```

```
MOV R0,#11D      ;TAMPILKAN HURUF Cm/Hg
MOV DPTR,#KAL14
LCALL PRINTSTRING1
```

```
MOV R1,#1D      ;TULIS SET PRESSURE
MOV DPTR,#KAL10
LCALL PRINTSTRING2
```

```
MOV A,SETZ      ;SET
LCALL OLAHDT
```



```
MOV R1,#10D      ;TULIS ERR PRESSURE-SETTING
MOV DPTR,#KAL11
LCALL PRINTSTRING2
```

```
MOV A,ERR_3      ;ERR
LCALL OLAHDT
```

```
RET
```

```
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;  PROCEDURE BACA DATA PIPA U
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
BACA_PIPA_U:
```

```
    MOV DPTR,#PB      ;baca data infra
    LCALL DLY_PPI      ;PPI Port B
    MOVX A,@DPTR
    MOV INFRA,A
    LCALL DLY_PPI
```

```
    MOV A,#10H        ;81 Cm/Hg
    CJNE A,INFRA,N_4
    MOV A,#81
    MOV TEKANAN,A
```

```
N_4:  MOV A,#08H      ;80 Cm/Hg
       CJNE A,INFRA,N_3
       MOV A,#80
       MOV TEKANAN,A
```

```
N_3:  MOV A,#04H      ;79 Cm/Hg
       CJNE A,INFRA,N_2
       MOV A,#79
       MOV TEKANAN,A
```

```
N_2:  MOV A,#02H      ;78 Cm/Hg
       CJNE A,INFRA,N_1
       MOV A,#78
       MOV TEKANAN,A
```

```
N_1:  MOV A,#01H      ;77 Cm/Hg
       CJNE A,INFRA,N_0
       MOV A,#77
       MOV TEKANAN,A
```

```
N_0:  MOV A,#76
```

MOV TEKANAN,A

RET

```
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;  PROCEDURE TAMPILKAN ERR & D ERR DI LCD
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ERR_DISP:
```

```
MOV R0,#1D      ;TULIS ERROR
MOV DPTR,#KAL15
LCALL PRINTSTRING1
```

```
MOV A,ERROR
LCALL OLAHDT
```

```
MOV R1,#1D      ;TULIS D ERR
MOV DPTR,#KAL16
LCALL PRINTSTRING2
```

```
MOV A,D_ERR
LCALL OLAHDT
```

RET

```
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;  PROCEDURE HITUNG ERROR DAN DELTA ERROR
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ERROR_SUHU:
```

; ----- hitung error suhu -----

```
CLR C
MOV A,SUHU
MOV B,SETX
CJNE A,SETX,NOT_EQ ;Suhu setting
MOV A,SUHU          ;A = #SETX
MOV B,SETX
SUBB A,B
MOV ERR_1,A
MOV A,#0
MOV SET_FUZ,A
LJMP LANJUT1
```

```
NOT_EQ: JC LEBIH_KECIL1 ;A < #SETX
MOV A,SUHU              ;A > #SETX
MOV B,SETX
```

```
SUBB A,B
MOV ERR_1,A
MOV A,#0
MOV SET_FUZ,A
AJMP LANJUT1
```

```
LEBIH_KECIL1:
    MOV A,SETX      ;A < #SETX
    MOV B,SUHU
    SUBB A,B
    ADD A,#1D
    MOV ERR_1,A
    MOV A,#08H
    MOV SET_FUZ,A
    AJMP LANJUT1
```

LANJUT1:

```
RET
```

; ----- hitung error kelembaban -----

```
ERROR_LEMBAB:
    CLR C
    MOV A,LEMBAB
    MOV B,SETY
    CJNE A,SETY,NOT_EQ2 ;Lembab setting
    MOV A,LEMBAB      ;A = #SETY
    MOV B,SETY
    SUBB A,B
    MOV ERR_2,A
    AJMP LANJUT2
```

```
NOT_EQ2:JC LEBIH_KECIL2      ;A < #SETY
    MOV A,LEMBAB      ;A > #SETY
    MOV B,SETY
    SUBB A,B
    MOV ERR_2,A
    AJMP LANJUT2
```

```
LEBIH_KECIL2:
    MOV A,SETY      ;A < #SETY
    MOV B,LEMBAB
    SUBB A,B
    ADD A,#1D
    MOV ERR_2,A
    AJMP LANJUT2
```

LANJUT2:

RET

;----- hitung error tekanan -----

ERROR_TEGANAN:

```
CLR C
MOV A,TEKANAN
MOV B,SETZ
CJNE A,SETZ,NOT_EQ3 ;tekanan setting
MOV A,TEKANAN ;A = #SETZ
MOV B,SETZ
SUBB A,B
MOV ERR_3,A
AJMP LANJUT3
```

```
NOT_EQ3:JC LEBIH_KECIL3 ;A < #SETZ
MOV A,TEKANAN ;A > #SETZ
MOV B,SETZ
SUBB A,B
MOV ERR_3,A
AJMP LANJUT3
```

LEBIH_KECIL3:

```
MOV A,SETZ ;A < #SETZ
MOV B,TEKANAN
SUBB A,B
ADD A,#1D
MOV ERR_3,A
AJMP LANJUT3
```

LANJUT3:

RET

;----- hitung error sistem -----

HITUNG_ERROR:

```
MOV A,ERR_1 ;error suhu
MOV B,ERR_2 ;error lembab
ADD A,B
MOV B,#2D
DIV AB
MOV B,ERR_3 ;error tekanan
ADD A,B
```



```
MOV B,#2D
DIV AB
MOV ERROR,A
```

```
RET
```

```
;----- error delay -----
```

```
UPDT:
```

```
MOV A,ERROR
MOV D_ERR,A
```

```
RET
```

```
;----- Kompensasi sistem -----
```

```
SETTING_POINT:
```

```
CLR A
MOV A,SET_FUZ
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI
```

```
RET
```

```
;----- Open Katup Stepper -----
```

```
STEPPER:
```

```
;----- PPI8255 -----
;----- P1.7 P1.6 P1.5 P1.4 -----
;----- open katup -----
```

```
MOV B,#5
```

```
BUKA:
```

```
LCALL DLY_PPI
CLR A
```

```
MOV A,#00010000B
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI3
```

```
MOV A,#00100000B
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI3
```

```
MOV A,#01000000B
MOV DPTR,#PC
MOVX @DPTR,A
```

LCALL DLY_PPI3

MOV A,#10000000B
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI3

DJNZ B,\$

MOV B,#5

TUTUP:

LCALL DLY_PPI
CLR A

MOV A,#10000000B
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI3

MOV A,#01000000B
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI3

MOV A,#00100000B
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI3

MOV A,#00010000B
MOV DPTR,#PC
MOVX @DPTR,A
LCALL DLY_PPI3

DJNZ B,\$

RET

;-----
DLY_PPI2:

MOV B,#0FFH
LOOP2:
MOV A,#0FFH
DJNZ A,\$
DJNZ B,LOOP2
RET

;-----
DLY_PPI3:

```
LCALL DLY_PPI2
LCALL DLY_PPI2
LCALL DLY_PPI2
LCALL DLY_PPI2
RET
```

;----- ubah setting -----

SET_SUHU:

LEPAS:

```
MOV R0,#1D      ;TULIS SET
MOV DPTR,#KAL17
LCALL PRINTSTRING1
```

```
MOV A,SETX      ;SET
LCALL OLAHDT
```

```
CLR A
MOV A,P1
ANL A,#02H
LCALL DLAY
CJNE A,#2,CEK1
```

TUNGGU1:

```
MOV A,P1
ANL A,#02H
LCALL DLAY
CJNE A,#0,TUNGGU1
```

```
MOV A,SETX
CJNE A,#80,TT01
MOV R1,#1D      ;LEVEL MAX
MOV DPTR,#KAL20
LCALL PRINTSTRING2
AJMP CEK2
```

TT01: INC A

```
MOV SETX,A
AJMP CEK2
```

CEK1:

```
MOV A,P1
ANL A,#04H
LCALL DLAY
CJNE A,#4,CEK2
```

TUNGGU2:

```
MOV A,P1
ANL A,#04H
LCALL DLAY
CJNE A,#0,TUNGGU2
```

```

MOV A,SETX
CJNE A,#40,TT02
MOV R1,#1D      ;LEVEL MIN
MOV DPTR,#KAL21
LCALL PRINTSTRING2
AJMP CEK2
TT02: DEC A
      MOV SETX,A

CEK2:
      MOV A,P1
      ANL A,#01H
      LCALL DLAY
      CJNE A,#1,LEPAS

      RET
;-----
SET_LEMBAB:
LEPAS2:
      MOV R0,#1D      ;TULIS SET
      MOV DPTR,#KAL18
      LCALL PRINTSTRING1

      MOV A,SETY      ;SET
      LCALL OLAHDT

      CLR A
      MOV A,P1
      ANL A,#02H
      LCALL DLAY
      CJNE A,#2,CEK3

TUNGGU3:
      MOV A,P1
      ANL A,#02H
      LCALL DLAY
      CJNE A,#0,TUNGGU3

      MOV A,SETY
      CJNE A,#90,TT03
      MOV R1,#1D      ;LEVEL MAX
      MOV DPTR,#KAL20
      LCALL PRINTSTRING2
      AJMP CEK4
TT03: INC A
      MOV SETY,A
      AJMP CEK4

```


CEK3:

```
MOV A,P1
ANL A,#04H
LCALL DLAY
CJNE A,#4,CEK4
```

TUNGGU4:

```
MOV A,P1
ANL A,#04H
LCALL DLAY
CJNE A,#0,TUNGGU4
```

```
MOV A,SETY
CJNE A,#80,TT04
MOV R1,#1D      ;LEVEL MIN
MOV DPTR,#KAL21
LCALL PRINTSTRING2
AJMP CEK4
```

```
TT04: DEC A
      MOV SETY,A
```

CEK4:

```
MOV A,P1
ANL A,#01H
LCALL DLAY
CJNE A,#1,LEPAS2
```

RET

;------

SET_TEGANAN:

LEPAS3:

```
MOV R0,#1D      ;TULIS SET
MOV DPTR,#KAL19
LCALL PRINTSTRING1
```

```
MOV A,SETZ      ;SET
LCALL OLAHDT
```

```
CLR A
MOV A,P1
ANL A,#02H
LCALL DLAY
CJNE A,#2,CEK5
```

TUNGGU5:

```
MOV A,P1
ANL A,#02H
LCALL DLAY
```

CJNE A,#0,TUNGGU5

MOV A,SETZ

CJNE A,#80,TT05

MOV R1,#1D ;LEVEL MAX

MOV DPTR,#KAL20

LCALL PRINTSTRING2

AJMP CEK6

TT05: INC A

MOV SETZ,A

AJMP CEK6

CEK5:

MOV A,P1

ANL A,#04H

LCALL DLAY

CJNE A,#4,CEK6

TUNGGU6:

MOV A,P1

ANL A,#04H

LCALL DLAY

CJNE A,#0,TUNGGU6

MOV A,SETZ

CJNE A,#76,TT06

MOV R1,#1D ;LEVEL MIN

MOV DPTR,#KAL21

LCALL PRINTSTRING2

AJMP CEK6

TT06: DEC A

MOV SETZ,A

CEK6:

MOV A,P1

ANL A,#01H

LCALL DLAY

CJNE A,#1,LEPAS3

RET

Features

- Complete, Single-Chip Fuzzy Logic Processor
- Flexible, Self-Adapting Control
- EEPROM Storage
- Four 8-Bit Analog Inputs
- Four 8-Bit Analog Outputs
- Six Types of Membership Functions
- 111 Fuzzy Variables
- Up to 50 Rules
- PC-Based Development System
- 18-Pin DIP or 20-Pin SOIC Package

Applications

- Smart Appliances
- Power and Battery Management
- Automotive
- Motor Control
- Communications

Description

The AL220 is an inexpensive, high-performance, stand-alone Fuzzy Logic Processor. The device performs Fuzzy Logic calculations directly in hardware. Because the AL220 is a dedicated controller, it offers superior ease of use, performance, features, and robustness in harsh operating environments.

Fuzzy Logic is a powerful processing methodology that easily accommodates imprecise input data and system nonlinearities for rapid development of robust control systems.

The methodology uses linguistic descriptions of systems, making it intuitive and simple to use. Fuzzy Logic can be used to inexpensively add intelligence to a wide variety of products to improve performance and features, and to increase efficiency.

Fuzzy Membership functions and rules are stored as parameters in EEPROM. The memory organization is flexible and efficiently adapts to the requirements of the application. The device stores 111 fuzzy variables, which are organized into as many as 50 rules.

The device supports six different types of membership functions to meet the requirements of any application. Membership functions are constant-slope and need only the type, width, and center specified.

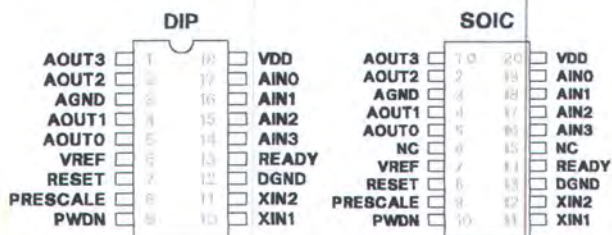
The AL220 offers Floating membership functions. The center and width of any membership function can be made to "Float" or vary dynamically. Floating membership functions can be used to measure derivatives, build timers, or adjust to drift in sensors.

Two methods of defuzzification are available: immediate and accumulate. The immediate mode drives an output with a specific value. The accumulate mode adds to the output's previous value.

Applications information is easily entered using the FL1000 Applications Expert development system running under Windows. Little knowledge of Fuzzy Logic is needed to use the devices or the system.

The AL220 is available in a 18-pin DIP or 20-pin SOIC package. The devices are ideal for a wide range of applications including appliances, motor control, automotive, and industrial systems.

Figure 1. Package Pin Assignments



Pin Description

Inputs

RESET An active-low signal that initializes the device. RESET should remain active for at least eight clock cycles to ensure proper operation. RESET can be driven by a power-up delayed circuit. Asserting RESET during operation causes the AL220 to enter its low-power mode.

AIN(3:0) Analog input data. Analog data is internally converted to 8-bit digital data. Unused inputs should be connected to ground.

CLKIN Clock input, driven by an external clock.

PRESCALE A logic level one puts the device into prescale mode while a zero causes normal operation. The pin can be grounded if prescale mode is never used. After RESET is deasserted, the PRESCALE pin must be held to a logic low for at least four clock cycles. Prescale operation is described later in this document.

Outputs

DAOUT(3:0) Analog output data. Eight-bit digital data is internally converted to an analog level.

READY After a reset, this pin signals that the device is starting to sample and process data.

VREF Filters the internal reference voltage. Connect to ground through a 0.1μF capacitor.

Table 1. Absolute Maximum Ratings $T_A = 25^{\circ}\text{C}$

Parameter	Min	Max	Units
V_{DD} to GND	-0.5	7.0	V
V_{SS} to GND	0	0	V
Digital Input	0	V_{DD}	V
Analog Input	0	V_{DD}	V
Power Dissipation		100	mW
Storage Temperature	-50	150	$^{\circ}\text{C}$

Table 2. Analog Conversion Specifications

Parameter	Value	Units
Resolution	1	BIT
Zero Code Error	$1 \pm$	LSB
Full Scale Error	$1 \pm$	LSB
Signal-to-Noise Ratio	45	dB min
Slew Rate, Tracking	1.6	V/mS max
Sampling Rate	10kHz	Per Channel

Table 3. Specifications and Recommended Operating Conditions

Parameter	Min	Norm	Max	Units
V_{DD} Supply Voltage	4.75	5.0	5.25	V
I_{DD} Supply Current				mA
I_{OL} Digital Output Low-Level Current			5	mA
I_{OH} Digital Output High-Level Current			-5	mA
F Clock Frequency	1		10	MHz
V_{IL} Digital Input Low-Level Voltage	0		0.8	V
V_{IH} Digital Input High-Level Voltage	3.5		V_{DD}	V
I_{IL} Digital Input Low-Level Current			-40	μA
I_{IH} Digital Input High-Level Current			1	μA
Z_{IN} Analog Input Impedance	100	150	250	kohm
V_{IN} Analog Input Voltage Range	0		$V_{DD}-0.5$	V
V_O Analog Output Voltage Range	$V_{SS}+0.5$		$V_{DD}-0.5$	V
I_O Analog Output Current	-5		5	mA
T_W Reset Pulse Width	100			mS
T_{SV} Reset Inactive Before Clock	10			mS
T_A Operating Ambient Temperature	0		70	$^{\circ}\text{C}$

The AL220

The device is a dedicated, stand-alone Fuzzy Logic controller. It performs all calculations in hardware and does not require software. The device is configured for a particular application through simple rules.

Device Architecture

A diagram of the AL220 is shown in Figure 2. The main elements are the Fuzzifier, Defuzzifier, and controller. The Fuzzifier converts input data into Fuzzy data. The Fuzzifier, in conjunction with the controller, evaluates fuzzy data by a user-defined set of rules that describes how the system is to be controlled. When the rules have been evaluated, the Defuzzifier assigns an action value to the appropriate output.

Developing a Fuzzy Logic System

To understand the operation of the device, you need to understand how to enter a Fuzzy Logic model and how to perform Fuzzy Logic calculations. The following sections explain the concepts underlying a Fuzzy system.

Membership Functions

Membership functions are separate categories used to define the range over which an input can vary. Membership functions are compared with input data to see where the data falls on them. They have names selected by the designer, such as Hot, Fast, or Tall, that classify data.

A household thermometer can be used to illustrate the concept of membership functions and show how Fuzzy Logic works like human reasoning. A person asked to divide the range of a thermometer according to comfort might designate temperatures as follows:

Below 60°F = Cold

60°F to 70°F = Cool

70°F to 75°F = Moderate

75°F to 85°F = Warm

Above 85°F = Hot

These divisions are an intuitive way for a person to consider temperature because they are based on the senses. A person could describe a room at 62°F as being cool. In Fuzzy Logic the five divisions are called membership functions and are represented in Figure 3. The membership functions can be separate, as shown, or they may overlap. That allows for data falling in the overlapped area to be a member of both functions. A temperature, for example, may be described as somewhat cool and somewhat cold.

Figure 2. AL220 Block Diagram

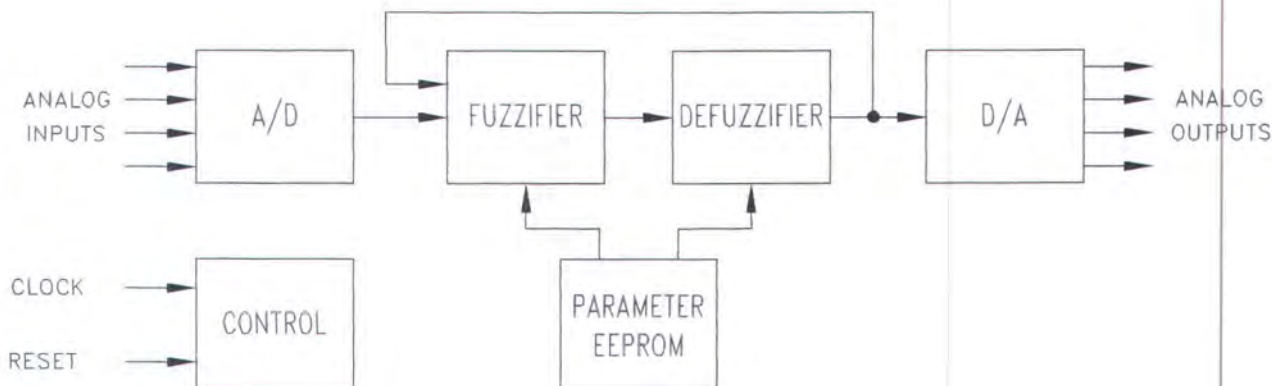
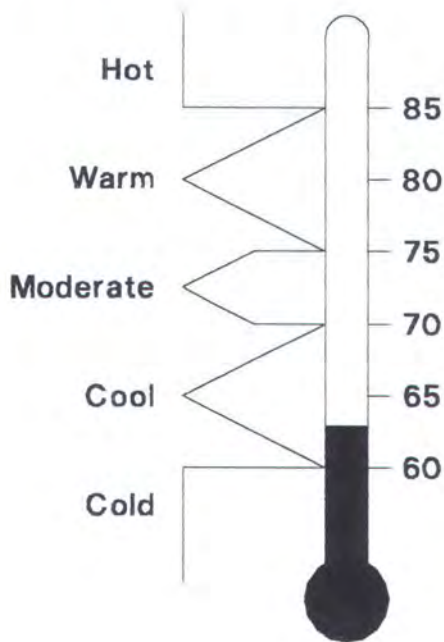


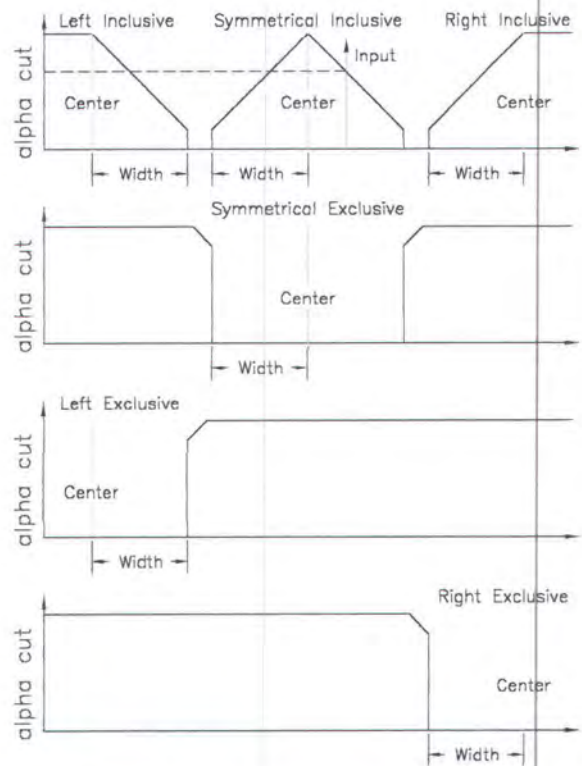
Figure 3. Temperature Membership Functions



The AL220 supports six different, constant slope, membership functions as shown in Figure 4. They consist of Left, Symmetrical, and Right Inclusive functions and their inverses, Exclusive functions. They are defined with a type that specifies a shape, with numerical values for the center and width.

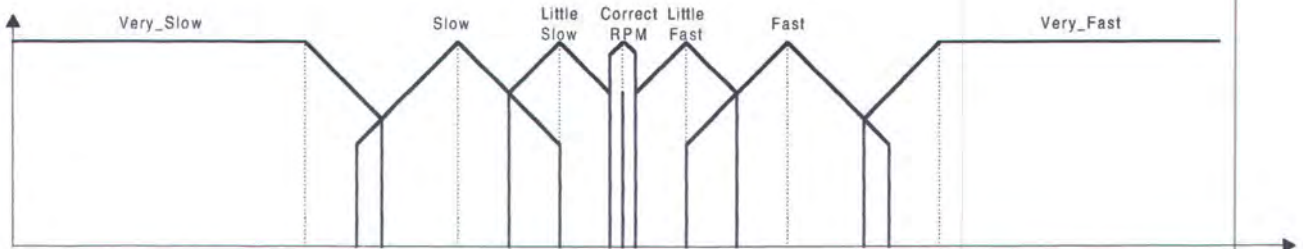
Careful selection of membership functions simplifies the description of many models. For example, single right or left inclusive membership functions are used to cover large ranges of values at the ends of the range of input variation. In the thermometer example, Cold would be represented by a left inclusive membership function and Hot by a right inclusive membership function.

Figure 4. Supported Membership Function Types



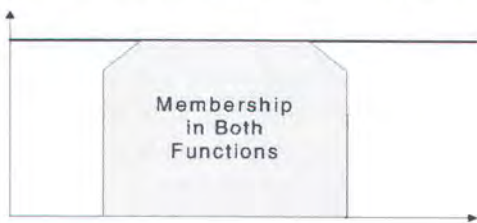
Precise control about the desired operating point can be of the narrow symmetrical inclusive type, which provides more precise control. Many motor control applications require such precision. An example of a mixture of different types and widths of membership functions used to monitor the velocity of a motor is given in Figure 5.

Figure 5. Velocity Membership Functions



Membership functions can be overlapped to create new shapes such as trapezoids as shown in Figure 6. The trapezoid is formed by overlaying a left and a right inclusive membership function. Inputs falling into the trapezoid are members of both functions.

Figure 6. Overlapped Membership Functions



Fuzzy Variables

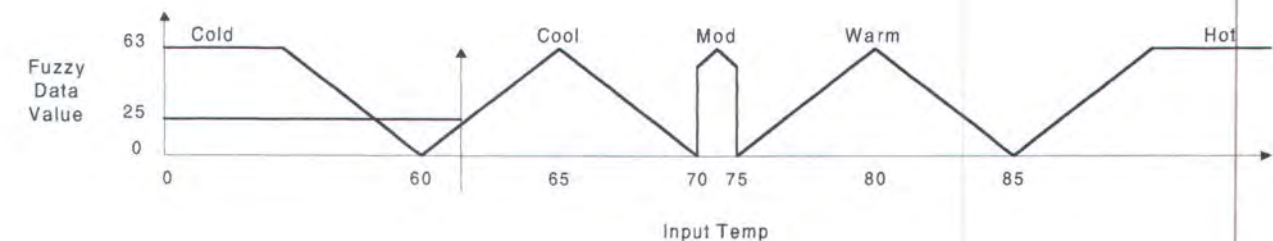
A fuzzy variable is a linguistic expression representing the association of an input value against the membership functions covering an axis. Fuzzy variables reference a membership function and an input variable. An example of a fuzzy variable is as follows:

If Temperature is Cool

In this example, 'Temperature' refers to an input and 'Cool' to a membership function.

The association is performed by the Fuzzifier. The result is a fuzzy variable value which represents the degree to which the input data matched the membership function. Fuzzy data values are numerical and range from 0 to 63 in the AL220. See Figure 7 for an example of a fuzzy variable evaluation.

Figure 7. Fuzzification of Input Temperature



Rules

A rule consists of one or more fuzzy variables and an action output value. Rules are used to tell the controller how to respond to changes in input data.

In the examples below, both rules contain two fuzzy variables. Rules are entered into the INSIGHT development system in the following format:

Output = -5 if Velocity is Fast and Acceleration is Positive

Output= +5 if Velocity is Little_Slow and Acceleration is Zero

In the first rule, the first fuzzy variable is the same as in the previous example and the second fuzzy variable is 'Acceleration is Positive.' The action '+5' is a numerical value that is applied to an output to slow down or speed up the motor.

Rule Evaluation

There are several methods for evaluating Fuzzy Logic rules. The AL220 evaluates rules using the two-step MAX-of-MINs technique.

In the first step (MIN), all the values for the fuzzy variables in a rule are compared; the lowest value represents the rule. In the second step (MAX), the values for the rules are compared; the rule with the highest value wins.

The way membership functions, fuzzy variables, and rules are defined and organized depends on the requirements of the application. You need to understand the physical properties of the system to be controlled before entering the Fuzzy model. If you are armed with that knowledge, however, entering a model is a straightforward process.

Floating Membership Function

A unique feature of the AL220 is the Floating membership function. As shown in Figure 8, floating membership functions have center and width values that vary dynamically. In ordinary membership functions the center and width are fixed values stored in memory. In a floating membership function, these values come from any of the inputs or outputs.

Figure 8. Floating Membership Functions



Any membership can be specified as floating during design entry. The floating membership function changes its center or width value as data from a selected input or output changes.

For example, two fuzzy variables with their membership functions described parenthetically could be defined conventionally and used in a rule as follows:

IN1 is small (0, 25, Symmetrical Inclusive)

IN2 is small (0, 25, Symmetrical Inclusive)

where the first number, zero, refers to the center and the second, 25, to the width of the membership function.

Output = +1 if IN1 is small and IN2 is small

where the fuzzy variable 'IN1 is small' compares input IN1 against the conventional membership function 'small.'

Floating membership functions make the same description more concisely in the following fuzzy variable and rule:

IN1 is small_difference (IN2, 25, Symmetrical Exclusive)

Output = +1 if IN1 is small_difference

In the fuzzy variable, the center of the membership function small_difference is defined by the value of IN2 stored in the input's latch.

In Fuzzification, an input is subtracted from the center of membership functions and the result inverted to measure how closely it matches the center value. When the device fuzzifies a membership function with a floating center, it subtracts one input from another.

Floating membership functions allow you to use a fuzzy variable that directly measures the difference between two inputs. The technique can be used, as in the example, to calibrate changes in a sensor over time.

The sensor's quiescent value is compared to a set voltage. Calibration rules check the degree of mismatch and store a correction value in an output. If the inputs are in calibration, the centers will match and the correction value is zero. Large mismatches will store large corrections.

The correction is used to adjust the floating center of a membership function in rules that process sensed data.

Floating membership functions can also be used to obtain the derivative of an input value by means similar to those described in the example above. A rule can reference an input as a floating action value causing it to be passed directly to an output.

During the next sample of the input, the output value selects the membership function center value, which has the effect of subtracting the previous input value from the current value. The difference can be referenced by a fuzzy variable in a rule.

An example of the use of an input/action value would be in measuring the acceleration of a motor. A rule that stores an input value into an output could be written as follows:

VALUE_TO - IN1 IF IN1 IS MUST_WIN (0, 0, Right Inclusive)

The rule references IN1 as an action value. The membership function MUST_WIN is a Right Inclusive type that begins at zero so that, regardless of the value of IN1, the rule must win and the value of IN1 is stored in the output.

A second rule calculates the derivative and adjusts the output that drives the motor.

ACCEL + if IN1 is VALUE_T1 (VALUE_TO, 25, Symmetrical Inclusive)

The rule determines whether the input's value at T1 is within 25 of its value at T0. In an actual application, there would be other membership functions to determine the polarity of the derivative and other rules to cover larger adjustments to larger variations.

The above examples of floating membership functions are straightforward. In an actual application, floating membership functions can be used extensively to save memory because they use fewer fuzzy variables and rules to detect differences between inputs than conventional functions do.

Device Operation

Processing data involves several steps. First, sampled analog data is converted to digital data. Next, the Fuzzifier compares the digital input data with the fuzzy variables to find a value for the fuzzy variable. The Fuzzifier also performs the MAX-of-MIN calculation to determine the winning rule. Finally, the Defuzzifier determines the winning rule's action value and sends it for conversion to an analog output or internal feedback.

Fuzzifier

The Fuzzifier compares input data with membership functions to calculate a Fuzzy variable value. When the MIN calculation has been performed on all the fuzzy variables in a rule, the value representing the rule is stored. When the MAX calculation has been performed on all the rules referencing an output, the winning rule's action value is passed to the Defuzzifier.

Rules are evaluated in the order they are entered. Any rule can reference any output. Outputs can be referenced repeatedly in a rule set.

When a rule or group of rules affecting an output has been entered and the next rule entered references another output, the Last Rule causes the output to be updated with the action value of the winning rule.

After processing rules that affect other outputs, the processor encounters another rule or group of rules referencing the same output, then it will update the output again. An output then may be updated as many times during a processing cycle as there are separate groups of rules referencing it.

As mentioned previously, input sampling is continuous. Analog output values are also updated continuously. During the course of a processing cycle, a fuzzy variable may use a data sample from the previous sample cycle or from the current cycle depending on where the input sample cycle is relative to the processing cycle. Should more than one group of rules reference the same input and output, then the output value may change more than once during a processing cycle based on different input data.

Of the order in which these outputs change, or the number of times an output changes during a processing cycle is insignificant, then the order of rules should be considered when they are entered.

Defuzzifier

The winning rule's action value and mode data are passed to the Defuzzifier block. Digital data from the

Defuzzifier is latched and converted to analog to drive the outputs or looped back internally.

If all the rules in a group referencing an output evaluate to zero, then the output will not change its value. If more than one rule evaluates to the same highest nonzero value, then the first of those rules entered will win and its action will determine the output.

Defuzzification Methods

Defuzzification causes the action value of the winning rule to drive an output. The device supports two methods of defuzzification, immediate and accumulate. Either of the two modes depicted in Figures 9 and 10 can be selected for a rule.

Immediate Mode functions like a lookup table, where the action value assigned to the winning rule during entry is applied to an output.

Immediate defuzzification is useful when the output value must be absolute or when large changes are required.

Accumulate Mode increments or decrements the existing output by the action value for the winning rule. The output is a function of the current action and the previous output.

Accumulate defuzzification can be used for subtle changes to outputs when the system under control is near a desired operating point. It is also useful for timing functions.

Figure 9. Immediate Defuzzification

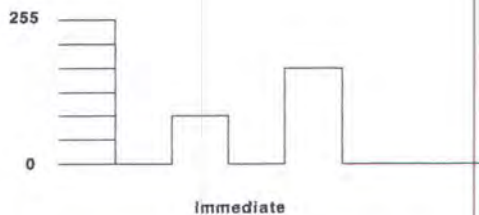


Figure 10. Accumulate Defuzzification



Memory Organization

The AL220 contains a 256-byte EEPROM for applications parameter storage. The last 32 bytes store fixed membership function Center and Width values. The remaining 224 bytes are organized as one or more rules with one or more fuzzy variables per rule.

Each rule requires two bytes, plus an additional two bytes for each fuzzy variable in that rule. A rule containing five fuzzy variables, for example, would use 12 bytes.

The memory is organized into three sections, defined as Rule/Fuzzy Variable storage, Center storage, and Width storage. Memory organization is illustrated in Table 4.

Table 4. Memory Organization

Dec Address	Hex Address	Function
0	00	Rules
<>	<>	<>
223	DF	Rules
224	E0	Centers
<>	<>	<>
239	EF	Centers
240	F0	Widths
<>	<>	<>
255	FF	Widths

Rule and fuzzy Variable Storage

Rules are organized as groups of one or more fuzzy variables. Each fuzzy variable is made up of two bytes, as described in Tables 5 and 6. The first byte is stored at even addresses and the second at odd addresses.

The types are divided into fields that control how data is processed. The three least significant bits of the even byte define either the membership functions type or whether the previous fuzzy variable was the last of the rule or the last fuzzy variable of the last rule referencing the output.

When the least significant field selects a membership function type, the five most significant bits select the input source for the fuzzy variable. The five-bit field is subdivided into a three-bit field that selects the input source from one of the four input pins or output latches. The remaining two most significant bits define whether the center and width of the membership function are floating or fixed.

The Type code Last Fuzzy variable (001) signals that the last fuzzy variable of the rule has been processed. When this occurs, only the two most significant bits (MSB) of the five-bit field are used. The MSB selects whether the action value comes from a fixed memory location or from an I/O latch. The next MSB specifies the output mode, immediate or accumulate.

The code (000) indicates Last Fuzzy variable of Last Rule. The two most significant bits are used as described in the paragraph above. In addition, the two bits above the Type Select field are used to select the output.

The second byte always occurs on an odd address, and contains the Center and Width address index fields if the previous byte specified a membership function type and fixed center and width value. If either the center or the width were specified as floating, then their respective nibble in the odd byte is used to select the input or output.

When the first byte's Type is Last fuzzy variable or Last Fuzzy variable of Last Rule and the action is fixed, the second byte contains the action value. If action is floating, then the odd byte selects the input or output that provides the action value.

Table 5. Command Byte (Even Addresses)

7	6	5	4	3	2	1	0
WF	CF	I/O CONT	I/O SELECT		TYPE 2-7		
AF	MODE				TYPE 1		
AF	MODE		OUTPUT SELECT		TYPE 0		

Key:

Type	210	
	0 0 0	Last Term of Last Rule of given output
	0 0 1	Last Term of Current Rule
	0 1 0	MF, Symmetrical, Inclusive
	0 1 1	MF, Symmetrical, Exclusive
	1 0 0	MF, Left, Inclusive
	1 0 1	MF, Left, Exclusive
	1 1 0	MF, Right, Inclusive
	1 1 1	MF, Right, Exclusive
I/O Select	43	
	0 0	I/O Port 0 as Input
	0 1	I/O Port 1 as Input
	1 0	I/O Port 2 as Input
	1 1	I/O Port 3 as Input
I/O Control	5	
	0	Select from Inputs
	1	Select from Outputs
Mode	6	
	0	Immediate, Output equals ACTION
	1	Accumulate, Output equals current output plus two's complement action (-128 to +127)
AF	7	
	0	Select Action from Select Byte (fixed)
	1	Select Action from I/O via Select Byte (float)
Output Select	43	
	0 0	ACTION from current RULE set to Output 0
	0 1	ACTION from current RULE set to Output 1
	1 0	ACTION from current RULE set to Output 2
	1 1	ACTION from current RULE set to Output 3
CF	6	
	0	Select Center from memory via Select Byte (fixed)
	1	Select Center from I/O via Select Byte (float)
WF	7	
	0	Select Width from Memory via Select Byte (fixed)
	1	Select Width from I/O via Select Byte (float)

Table 6. Select Byte (Odd Addresses)

7	6	5	4	3	2	1	0
CENTER SELECT				WIDTH SELECT			TYPE=2-7,CF or WF=0 (FIXED)
	I/O CONT	I/O SELECT CENTER			I/O CONT	I/O SELECT WIDTH	TYPE=2-7, CF or WF=1 (FLOAT)
ACTION							TYPE=0-1, AF=0 (FIXED)
					I/O CONT	I/O SELECT ACTION	TYPE=0-1, AF=0 (FLOAT)

Key:

Width Select	(3:0)	Used as Address Index (EO-EF) for Fixed 6-bit WIDTH Value when Type =2-7 and WF=0
Center Select	(7:4)	Used as Address Index (FO-FF) for Fixed 8-CENTER Value when Type =2-7 and CF=0
I/O Select Width	10	
	00	I/O Port 0 as Width (Type=2-7 and WF=1)
	01	I/O Port 1 as Width (Type=2-7 and WF=1)
	10	I/O Port 2 as Width (Type=2-7 and WF=1)
	11	I/O Port 3 as Width (Type=2-7 and WF=1)
I/O Control	2	
	0	Select from Inputs (Type=2-7 and WF=1)
	1	Select from Outputs (Type=2-7 and WF=1)
I/O Select Center	54	
	00	I/O Port 0 as Input (Type=2-7 and WF=1)
	01	I/O Port 1 as Input (Type=2-7 and CF=1)
	10	I/O Port 2 as Input (Type =2-7 and CF=1)
	11	I/O Port 3 as Input (Type=2-7 and CF=1)
I/O Control	6	
	0	Select from Inputs (Type=2-7 and CF=1)
	1	Select from Outputs (Type=2-7 and CF=1)
ACTION	7-0	8-bit Action value to be applied to an output due to a winning Last Term of a Rule (TYPE=1) or Last Term of Last Rule of a given Output (Type=0), and AF=0 (Fixed)
I/O Select Action	10	
	00	I/O Port 0 as Action (Type=1-0 and AF=1)
	01	I/O Port 1 as Action (Type=1-0 and AF=1)
	10	I/O Port 2 as Action (Type=1-0 and AF=1)
	11	I/O Port 3 as Action (Type=1-0 and AF=1)
I/O Control	2	
	0	Select from Inputs (Type=1-0 and AF=1)
	1	Select from Outputs (Type=1-0 and AF=1)

Timing

Figure 12 illustrates timing for the AL220. The three architectural blocks that impact timing include the multiplexed input A/D converter, the Fuzzy Processor, and the multiplexed output D/A converter.

Processing speed is a function of both the clock rate and the number of clocks (1024) required to complete data sampling and processing cycles. The clock maximum rate is 10 MHz and the minimum is 1mhz.

Operating Timing

Reset When the RESET pin is active, all the latches are cleared, the digital outputs are logic low, and the analog outputs hold at the level they were at prior to the assertion of reset. If RESET is active for one hundred clocks or more, the analog inputs will be zero when sampling resumes. If RESET is active for less than one hundred clocks, there may be some residual of the last sampled data still present on the analog inputs when sampling resumes. When RESET is deactivated the device begins sampling inputs during the first 1024 clock cycle.

Input Conversion Input analog values are converted to digital data and latched internally in successive periods of 256 clocks each. A total of 1024 clock cycles are required to convert all four inputs after which the conversion process repeats. At the maximum clock, the sample rate for each input is 10 KHz, or 100 microseconds.

Processor Timing The first 1024 clock processing cycle begins after the first input conversion cycle has completed. Processing cycles consist of 1024 clock cycles regardless of the number of fuzzy variables and rules used.

Fuzzy variable and rule evaluations require four clocks each. For example, a rule with two fuzzy variables would require 12 clocks to process. During a processing cycle either a fuzzy variable or rule is being processed each four-clock period, except for a 64-clock latency period at the end of the processing cycle.

Internal Loopback Delay

When data in the output latch is internally looped back as inputs, they lag behind the analog inputs by the 1024 clocks of the initial sampling cycle. After that, as the output latches are updated during processing the data feedback is used as inputs.

Prescaled Operation

The device contains a loadable prescale counter that allows it to be inactive for periods of time. The feature is used to vary the rates of sampling and processing. The last location in the memory, which normally stores fixed membership function width data, may instead store a value to be loaded into the counter. The PRESCALE pin selects normal or prescaled operation.

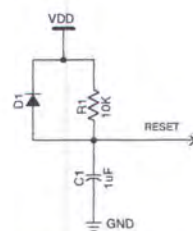
In prescale mode, the processor is inactive for periods of 1024 clocks after which the counter is incremented. When the counter reaches a value of FF, the processor is activated for a single 1024 clock period to perform Fuzzy computations and the counter is loaded again.

The Prescale pin can be connected to the Ready pin when prescale operation is desired.

Powerup Reset and Clock

Figure 11 shows a typical power up reset circuit.

Figure 11. Powerup Reset Circuit



The Clock can be derived from Figure 12.

Figure 12. Clock Circuits

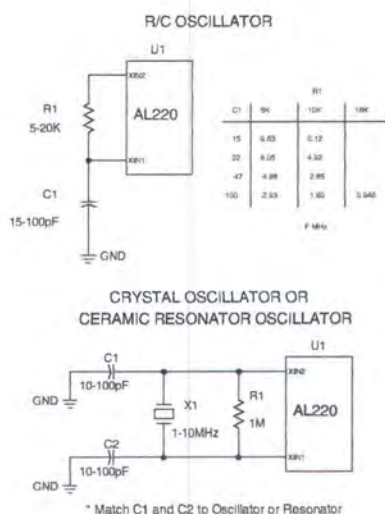


Figure 13. Timing Considerations

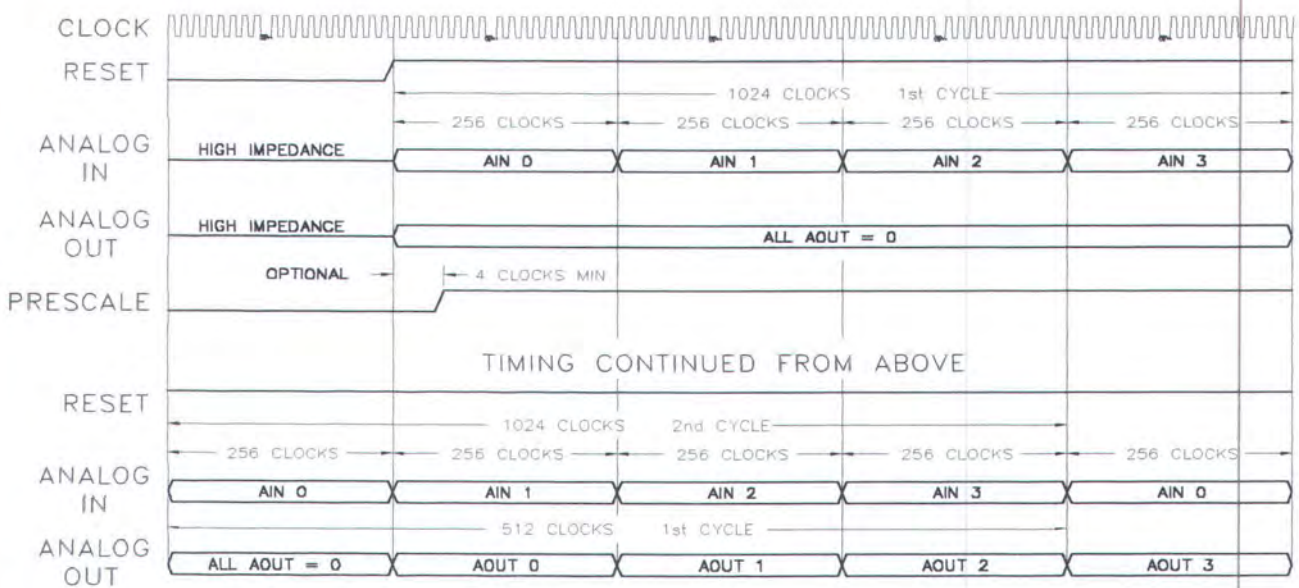
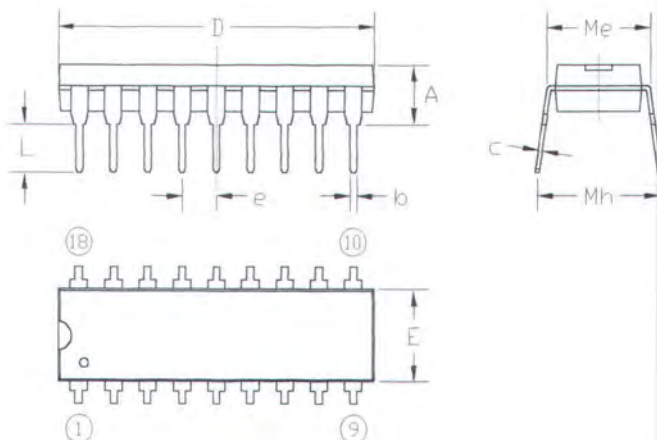
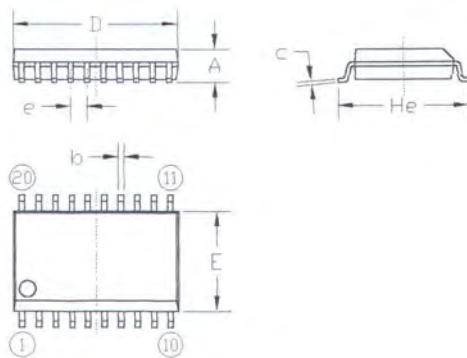


Figure 14. DIP Package Mechanical Details



UNIT	A	b	c	D	E	e	L	Me	Mh
	4.06	1.63	0.36	23.50	6.48		3.51	8.13	10.03
mm						2.54			
		1.14	0.25	23.24	6.22		3.05	7.62	7.62
	0.160	0.064	0.014	0.925	0.255		0.138	0.32	0.295
inches						0.100			
		0.045	0.010	0.915	0.245		0.120	0.30	0.300

Figure 15. SOIC Package Mechanical Details



UNIT	A	b	c	D	E	e	He
	2.65	0.49	0.32	13.0	7.6		10.65
mm						1.27	
		0.36	0.23	12.6	7.4		10.00
	0.10	0.019	0.013	0.51	0.30		0.42
inches						0.050	
		0.014	0.009	0.49	0.29		0.39

asiConcepts reserves the right to make changes in this document without notice

asiConcepts

12 W. First St. Suite 201
 Sanford, FL 32771
 407-322-5608
 Fax: 407-322-5609

Printed in U.S.A.

RIWAYAT HIDUP



SETIAWARDHANA dilahirkan di Surabaya pada tanggal 24 Agustus 1977. Putra pertama dari satu bersaudara dari :

Ayah : Harijanto Pontjoresi , SH.

Ibu : R.A. Poernomo Retno Soesilo

Yang bertempat tinggal di Jalan Barata Jaya III / 40
Surabaya

Terdaftar sebagai mahasiswa Jurusan Teknik Elektro, Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember pada tahun 1995 melalui jalur UMPTN dengan nomor registrasi pokok 2295.100.029.

Pendidikan yang telah ditempuh :

1. SD Negeri Barata Jaya I No 202 , tahun 1983 - 1989
2. SMP Negeri 6 Surabaya , tahun 1989 - 1992
3. SMA Negeri 16 Surabaya , tahun 1992 – 1995
4. Jurusan Teknik Elektro FTI – ITS Surabaya , tahun 1995 – sekarang

Selama menjadi mahasiswa aktif sebagai Koordinator Laboratorium Elektronika , Koordinator Praktikum Elektronika, Asisten Praktikum Rangkaian Listrik , Asisten Praktikum Elektronika, dan Asisten Praktikum Elektronika Lanjutan II di Bidang Studi Elektronika Jurusan Teknik Elektro FTI – ITS Surabaya.